(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 July 2021 (29.07.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/146933 A1

(51) International Patent Classification:
H04N 19/117 (2014.01) *H04N 19/82* (2014.01)
H04N 19/86 (2014.01)

(21) International Application Number:
PCT/CN2020/073563

(22) International Filing Date:
21 January 2020 (21.01.2020)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **ALIBABA GROUP HOLDING LIMITED** [—/CN]; Fourth Floor, One Capital Place, P.O. Box 847, George Town, Grand Cayman (KY).

(72) Inventors: **CHANG, Tsuishan**; Building 2, Wanmei Modern Commercial Center, Zhuantang, Xihu District, Hangzhou, Zhejiang 310024 (CN). **SUN, Yuchen**; 938 110th Ave NE B506, Bellevue, WA 98004 (US). **LOU, Jian**; 10270 Ne 12th St F305, Bellevue, WA 98004 (US).

(74) Agent: **BEIJING TSINGYUANHUI INTELLECTUAL PROPERTY LAW FIRM**; Room 101, 1st Floor, Building 1, No. C-18, Zhichun Road, Haidian District, Beijing 100190 (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*

(54) Title: NEXT-GENERATION LOOP FILTER IMPLEMENTATIONS FOR ADAPTIVE RESOLUTION VIDEO CODING

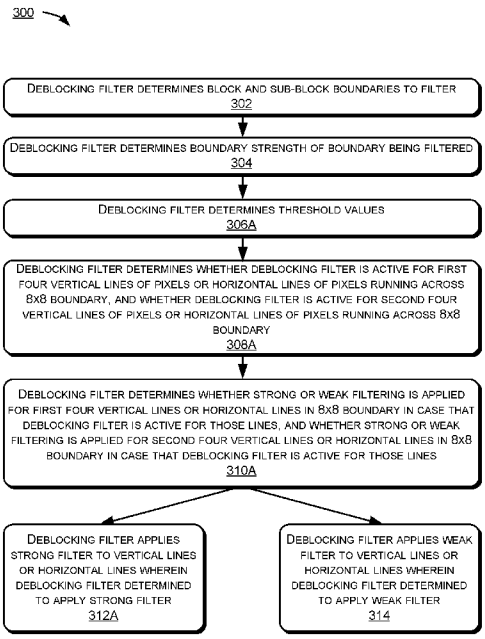

FIG. 3A

(57) Abstract: Systems and methods are provided for implementing inter-coded resolution-adaptive video coding supported by multiple in-loop filters, restoring high-frequency loss that occurs when a picture is down-sampled and subsequently up-sampled, and improving image quality during a resolution-adaptive video coding process. The methods and systems described herein provide a deblocking filter which takes resolution differences between frames undergoing motion prediction into account in determining filter strength, with further modifications for the next-generation video codec specification VVC. The deblocking filter may apply a strong filter or a weak filter in cases where a first reference frame referenced in motion prediction of a block adjacent to the block boundary has a resolution different from that of a second reference frame referenced in motion prediction of a block adjacent to the block boundary, that of a second reference frame referenced in motion prediction of the current frame, or that of the current frame.

WO 2021/146933 A1

NEXT-GENERATION LOOP FILTER IMPLEMENTATIONS FOR ADAPTIVE RESOLUTION VIDEO CODING

BACKGROUND

[0001] In conventional video coding formats, such as the H.264/AVC (Advanced Video Coding) and H.265/HEVC (High Efficiency Video Coding) standards, video frames in a sequence have their size and resolution recorded at the sequence-level in a header. Thus, in order to change frame resolution, a new video sequence must be generated, starting with an intra-coded frame, which carries significantly larger bandwidth costs to transmit than inter-coded frames. Consequently, although it is desirable to adaptively transmit a down-sampled, low resolution video over a network when network bandwidth becomes low, reduced or throttled, it is difficult to realize bandwidth savings while using conventional video coding formats, because the bandwidth costs of adaptively down-sampling offset the bandwidth gains.

[0002] Research has been conducted into supporting resolution changing while transmitting inter-coded frames. However, such developments impose new requirements on not just the coding and decoding parts of the coding loop, but also further interdependent processing thereafter. For example, one or more in-loop filters are conventionally applied to a frame after it has been reconstructed by an in-loop encoder and/or after it has been decoded by an in-loop decoder.

[0003] According to both the H.264/AVC and H.265/HEVC standards, a deblocking filter is applied to a reconstructed frame output by an in-loop encoder. Block-based coding algorithms as established by these standards tend to produce artifacts known as "blocking," which may be ameliorated by a deblocking filter. Furthermore, according to the H.265/HEVC standard, a sample adaptive offset (SAO) filter may be applied to the reconstructed frame output by the deblocking filter. In the development of the next-generation video codec specification, VVC, a third in-loop filter, the adaptive loop filter (ALF) is applied to the reconstructed frame output by the SAO filter.

[0004] The current implementations of these filters do not take resolution changes into account, and so new techniques are required in order to cause each of these filters

to behave correctly when the reconstructed frame has a resolution different from other frames in a buffer. It is also desirable to implement these techniques based on practical expectations regarding different severity of blocking artifacts in different scenarios.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The detailed description is set forth with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different figures indicates similar or identical items or features.

[0006] FIGS. 1A and 1B illustrate an example block diagram of a video encoding process and a video decoding process according to example embodiments of the present disclosure.

[0007] FIGS. 2A through 2D illustrate coding loop flows including different arrangements of an up-sampler and multiple in-loop filters according to example embodiments of the present disclosure.

[0008] FIGS. 3A and 3B illustrates deblocking methods performed by a deblocking filter according to the HEVC and VVC specifications and to example embodiments of the present disclosure.

[0009] FIGS. 4A and 4B illustrate flowcharts of deblocking filter logic according to example embodiments of the present disclosure.

[0010] FIG. 5 illustrates a deblocking filter computing bS values by reference.

[0011] FIGS. 6A and 6B illustrates determining whether the deblocking filter is active for a first four horizontal lines and a second four horizontal lines in an 8x8 pixel boundary according to example embodiments of the present disclosure.

[0012] FIG. 7 illustrates an example flowchart of a sample adaptive offset (SAO) filter method according to example embodiments of the present disclosure.

[0013] FIG. 8A illustrates an edge pattern made up of pixels including the current pixel *p* and two neighbor pixels at a 0-degree angle. FIG. 8B illustrates an edge pattern made up of pixels including the current pixel *p* and two neighbor pixels at a 90-degree angle. FIG. 8C illustrates an edge pattern made up of pixels including the current pixel *p* and two neighbor pixels at a 135-degree angle. FIG. 8D illustrates an edge pattern

made up of pixels including the current pixel *p* and two neighbor pixels at a 45-degree angle.

[0014] FIG. 9A illustrates an example flowchart of an adaptive loop filter (ALF) method according to example embodiments of the present disclosure.

[0015] FIG. 9B illustrates shapes of ALF filters.

[0016] FIGS. 9C through 9F illustrate subsampling over a sub-block for calculating vertical, horizontal, and two diagonal gradient values of a sub-block of the luma coding tree block (CTB).

[0017] FIG. 10 illustrates an example system for implementing the processes and methods described above for implementing resolution-adaptive video coding in deblocking filters.

[0018] FIG. 11 illustrates an example system for implementing the processes and methods described above for implementing resolution-adaptive video coding in SAO filters.

[0019] FIG. 12 illustrates an example system for implementing the processes and methods described above for implementing resolution-adaptive video coding in ALF.

DETAILED DESCRIPTION

[0020] Systems and methods discussed herein are directed to integrate inter-frame adaptive resolution change with a video coding loop, and more specifically to in-loop filter methods which improve inter-frame adaptive resolution change and process reconstructed frames output by inter-frame adaptive resolution change.

[0021] According to example embodiments of the present disclosure implemented to be compatible with AVC standards, a frame may be subdivided into macroblocks (MBs) each having dimensions of 16x16 pixels, which may be further subdivided into partitions. According to example embodiments of the present disclosure implemented to be compatible with the HEVC standard, a frame may be subdivided into coding tree units (CTUs), the luma and chroma components of which may be further subdivided into coding tree blocks (CTBs) which are further subdivided into coding units (CUs). According to example embodiments of the present disclosure implemented as other standards, a frame may be subdivided into units of NxN pixels, which may then be

further subdivided into subunits. Each of these largest subdivided units of a frame may generally be referred to as a "block" for the purpose of this disclosure.

[0022] According to example embodiments of the present disclosure, a block may be subdivided into partitions having dimensions in multiples of 4x4 pixels. For example, a partition of a block may have dimensions of 8x4 pixels, 4x8 pixels, 8x8 pixels, 16x8 pixels, or 8x16 pixels.

[0023] According to example embodiments of the present disclosure, motion prediction coding formats may refer to data formats wherein frames are encoded with motion vector information and prediction information of a frame by the inclusion of one or more references to motion information and prediction units (PUs) of one or more other frames. Motion information may refer to data describing motion of a block structure of a frame or a unit or subunit thereof, such as motion vectors and references to blocks of a current frame or of another frame. PUs may refer to a unit or multiple subunits corresponding to a block structure among multiple block structures of a frame, such as an MB or a CTU, wherein blocks are partitioned based on the frame data and are coded according to established video codecs. Motion information corresponding to a PU may describe motion prediction as encoded by any motion vector coding tool, including, but not limited to, those described herein.

[0024] Likewise, frames may be encoded with transform information by the inclusion of one or more transformation units (TUs). Transform information may refer to coefficients representing one of several spatial transformations, such as a diagonal flip, a vertical flip, or a rotation, which may be applied to a sub-block.

[0025] Sub-blocks of CUs such as PUs and TUs may be arranged in any combination of sub-block dimensions as described above. A CU may be subdivided into a residual quadtree (RQT), a hierarchical structure of TUs. The RQT provides an order for motion prediction and residual coding over sub-blocks of each level and recursively down each level of the RQT.

[0026] An encoder according to motion prediction coding may obtain a current frame of a bitstream and derive a reconstructed frame (a "reconstructed frame"). Blocks of a reconstructed frame may be intra-coded or inter-coded.

[0027] A CTU may include as components a luma CTB and a chroma CTB. A luma CTB of the CTU may be divided into luma sub-blocks. A chroma CTB of the CTU may be divided into chroma sub-blocks, wherein each chroma sub-block may have four neighboring luma sub-blocks. For example, a neighboring luma sub-block may be a luma sub-block below, left of, right of, or above the chroma sub-block.

[0028] Luma and chroma sub-blocks may be partitioned in accordance with PUs and TUs as described above – that is, partitioned into sub-blocks having dimensions in multiples of 4x4 pixels.

[0029] FIGS. 1A and 1B illustrate an example block diagram of a video encoding process 100 and a video decoding process 118 according to an example embodiment of the present disclosure.

[0030] In a video encoding process 100, a picture from a video source 102 may be encoded to generate a reconstructed frame, and output the reconstructed frame at a destination such as a reference frame buffer 104 or a transmission buffer 116. The picture may be input into a coding loop, which may include the steps of inputting the picture into a first in-loop up-sampler or down-sampler 106, generating an up-sampled or down-sampled picture, inputting the up-sampled or down-sampled picture into a video encoder 108, generating a reconstructed frame based on a previous reconstructed frame of the reference frame buffer 104, inputting the reconstructed frame into one or more in-loop filters 110, and outputting the reconstructed frame from the loop, which may include, or may not include: inputting the reconstructed frame into a second up-sampler or down-sampler 114, generating an up-sampled or down-sampled reconstructed frame, and outputting the up-sampled or down-sampled reconstructed frame into the reference frame buffer 104 or into a transmission buffer 116 to be transmitted to a bitstream.

[0031] In a video decoding process 118, a coded frame is obtained from a source such as a bitstream 120. According to example embodiments of the present disclosure, given a current frame having position N in the bitstream 120, a previous frame having position N–1 in the bitstream 120 may have a resolution larger than or smaller than a resolution of current frame, and a next frame having position N+1 in the bitstream 120

may have a resolution larger than or smaller than the resolution of the current frame. The current frame may be input into a coding loop, which may include the steps of inputting the current frame into a video decoder 122, inputting the current frame into one or more in-loop filters 124, inputting the current frame into a third in-loop up-sampler or down-sampler 128, generating an up-sampled or down-sampled reconstructed frame, and outputting the up-sampled or down-sampled reconstructed frame into the reference frame buffer 104. Alternatively, the current frame may be output from the loop, which may include outputting the up-sampled or down-sampled reconstructed frame into a display buffer (not illustrated).

[0032] According to example embodiments of the present disclosure, the video encoder 108 and the video decoder 122 may each implement a motion prediction coding format, including, but not limited to, those coding formats described herein. Generating a reconstructed frame based on a previous reconstructed frame of the reference frame buffer 104 may include inter-coded motion prediction as described herein, wherein the previous reconstructed frame may be an up-sampled or down-sampled reconstructed frame output by the in-loop up-sampler or down-sampler 114/128, and the previous reconstructed frame serves as a reference picture in inter-coded motion prediction as described herein.

[0033] According to example embodiments of the present disclosure, a first up-sampler or down-sampler 106, a second up-sampler or down-sampler 114, and a third up-sampler or down-sampler 128 may each implement an up-sampling or down-sampling algorithm suitable for respectively at least up-sampling or down-sampling coded pixel information of a frame coded in a motion prediction coding format. A first up-sampler or down-sampler 106, a second up-sampler or down-sampler 114, and a third up-sampler or down-sampler 128 may each implement an up-sampling or down-sampling algorithm further suitable for respectively upscaling and downscaling motion information such as motion vectors.

[0034] A frame serving as a reference picture in generating a reconstructed frame for the current frame, such as the previous reconstructed frame, may therefore be up-sampled or down-sampled in accordance with the resolution of the current frame

relative to the resolutions of the previous frame and of the next frame. For example, the frame serving as the reference picture may be up-sampled in the case that the current frame has a resolution larger than the resolutions of either or both the previous frame and the next frame. The frame serving as the reference picture may be down-sampled in the case that the current frame has a resolution smaller than either or both the previous frame and the next frame.

[0035] In light of the video coding process 100 as described above, a frame may be, for example, down-sampled in the encoding process of the coding loop by a down-sampler 106, and then the frame may be up-sampled by an up-sampler 114 or an up-sampler 128 and output into a reference frame buffer 104. A frame being down-sampled causes quality loss to the content of the frame, particularly high-frequency loss – for example, loss of sharp edges or fine patterns in the frame. This loss is not restored upon the subsequent up-sampling of the frame, resulting in a frame at its own original frame resolution but lacking high frequency detail. The use of a frame suffering from high-frequency loss as a reference frame in a reference frame buffer as described above leads to poor results for motion prediction of subsequent frames referencing the reference frame.

[0036] According to video coding standards such as H.264/AVC, H.265/HEVC, VVC, and the like, several types of in-loop filters may be conventionally applied to a reconstructed frame output from an encoder or a decoder. Given the implementation of an up-sampler or down-sampler in a coding loop, in-loop filters may, at a further level of granularity, be applied before or applied after an up-sampler.

[0037] FIGS. 2A through 2D illustrate coding loop flows including different arrangements of an up-sampler and multiple in-loop filters, including, for example, a deblocking filter, a SAO filter, and ALF.

[0038] According to example embodiments of the present disclosure, FIGS. 2A through 2D should be understood as illustrating receiving a frame that has been down-sampled during a video encoding process, as illustrated by FIG. 1A, and is to be up-sampled and output into a reference frame buffer. However, the frame may be received during the video encoding process 100 from a video encoder 108 as illustrated by FIG.

1A or may be received during the video decoding process 118 from a video decoder 122 as illustrated by FIG. 1B.

[0039] As illustrated by FIG. 2A, the up-sampler 211 may receive a down-sampled frame output by a video encoder 108 in the case of a video encoding process 100 as illustrated by FIG. 1A, or a down-sampled frame output by a video decoder 122 in the case of a video decoding process 118 as illustrated by FIG. 1B. The up-sampler 211 may up-sample the down-sampled frame and output the frame to the filters, including the deblocking filter 212, the SAO filter 213, and the ALF 214. This enables the encoder to analyze gradient, activity, and similar information of the frame, and utilize in-loop filter tools and parameters and coefficients thereof to evaluate image quality; furthermore, the encoder can transmit optimized parameters and/or coefficients of a deblocking filter, a SAO filter, and ALF to the decoder to enhance accuracy, objective quality, and subjective quality of reconstructed signals.

[0040] A deblocking filter 212 may filter a frame on a per-boundary per-CU basis in a coding order among CUs of the frame, such as a raster scan order wherein a first-coded CU is an uppermost and leftmost CU of the frame, according to video encoding standards. Within a frame, the deblocking filter 212 may filter both CUs of a luma CTB and a chroma CTB of the frame.

[0041] FIGS. 3A and 3B illustrates a deblocking method 300 performed by a deblocking filter 212. For illustrative purposes, and without limitation thereto, the deblocking method 300 may be performed according to the HEVC specification or according to the VVC specification. Certain differences between the HEVC specification implementation and the VVC specification implementation thereof shall be noted herein for ease of understanding with reference to FIG. 3A and FIG. 3B, respectively, though this shall not be understood as being a comprehensive accounting of all such differences.

[0042] At step 302, the deblocking filter 212 determines block and sub-block boundaries to filter. Within a block, according to the HEVC specification, the deblocking filter 212 may filter NxN pixel boundaries of sub-blocks of the CU. For example, the deblocking filter 212 may filter PU boundaries based on differences

between motion vectors and reference frames of neighboring prediction sub-blocks. That is, the deblocking filter 212 may filter PU boundaries in the event that the difference in at least one motion vector component between blocks on different sides of the boundary is greater than or equal to a threshold of one sampled pixel. Additionally, the deblocking filter 212 may filter TU boundaries in the event that coefficients sampled from pixels of a transform sub-block on either side of the boundary are non-zero. As a consequence, the deblocking filter 212 also filters those boundaries of the CU itself which coincide with outer PU and TU boundaries. However, according to the HEVC specification, sub-block boundaries filtered by the deblocking filter may be at least 8x8 pixels in dimensions, such that boundaries of 4x4 sub-blocks, such as TU sub-blocks representing 4x4 transforms, are not filtered by the deblocking filter 212, reducing filter complexity. Particularly, when a PU has dimensions of 2NxN pixels and N is greater than 4 pixels and the PU is at RQT depth 1 (that is, the first level of the RQT having the largest sub-block dimensions among levels), the deblocking filter 212 may filter PU boundaries between PUs in addition to outer PU boundaries and may filter TU boundaries at an 8x8 pixel grid in the RQT level.

[0043] Alternately, according to VVC specifications, sub-block boundaries filtered by the deblocking filter 212 may also be 4x4 pixels in dimensions where a boundary to filter is a boundary of a luma CTB, including CU boundaries and transform sub-block boundaries. Transform sub-block boundaries may include, for example, transform unit boundaries included by sub-block transform ("SBT") and intra sub-partitioning ("ISP") modes, and transforms due to implicit split of large CUs. Sub-block boundaries filtered by the deblocking filter 212 may still be 8x8 pixels in dimensions where a boundary to filter is a prediction sub-block boundary. Prediction sub-block boundaries may include, for example, prediction unit boundaries introduced by spatio-temporal motion vector prediction ("STMVP"), sub-block temporal motion vector prediction ("SbTMVP"), and affine motion prediction modes.

[0044] As in the implementations pertaining to TU boundaries according to HEVC specifications, the deblocking filter 212 may filter SBT and ISP boundaries in the event that coefficients sampled from pixels of a transform sub-block on either side of the

boundary are non-zero. Moreover, to facilitate concurrent computation of the deblocking filter 212 by parallel computing threads, in the event that the filtered boundary is also part of a STMVP, SbTMVP, or affine motion prediction sub-block, the deblocking filter 212 filters at most five samples on one side of the filtered boundary.

[0045] As in the implementations pertaining to PU boundaries according to HEVC specifications, the deblocking filter 212 may filter STMVP, SbTMVP, and affine motion prediction boundaries based on differences between motion vectors and reference frames of neighboring prediction sub-blocks. That is, the deblocking filter 212 may filter PU boundaries in the event that the difference in at least one motion vector component between blocks on different sides of the boundary is greater than or equal to a threshold of half a sampled luma pixel. Thus, blocking artifacts originating from boundaries between inter prediction blocks having only a small difference in motion vectors are filtered. Moreover, to facilitate concurrent computation of the deblocking filter 212 by parallel computing threads, in the case that four pixels are sampled between the filtered boundary and a transform block boundary, the filtered boundary is filtered by at most one sampled pixel on each side; in the case that eight pixels are sampled between the filtered boundary and a transform block boundary, the filtered boundary is filtered by at most two sampled pixels on each side; and in the case that any other number of pixels are sampled between the filtered boundary and a transform block boundary, the filtered boundary is filtered by at most three sampled pixels on each side.

[0046] Differences between the HEVC and VCC implementations are further described subsequently with reference to at least the alternate steps 312A and 312B.

[0047] Within an NxN pixel grid, a deblocking filter 212 may first filter vertical edges of a CU to perform horizontal filtering, and then filter horizontal edges of a CU to perform vertical filtering.

[0048] A deblocking filter 212 filtering a boundary of a current CU of a frame may reference a block P and a block Q adjacent to the boundary on both sides. While the deblocking filter 212 performs horizontal filtering, a block P may be a block left of the boundary and a block Q may be a block right of the boundary. While the deblocking

filter 212 performs vertical filtering, a block P may be a block above the boundary and a block Q may be a block below the boundary.

[0049] The deblocking filter 212 may identify a frame as having been down-sampled by determining that an inter-coded block P or block Q has a reference frame having a resolution different from a resolution of the current frame.

[0050] At step 304, the deblocking filter 212 determines a boundary strength (bS) of a boundary being filtered. A bS value may determine a strength of deblocking filtering to be applied by the deblocking filter to a boundary. A bS value may be 0, 1, or 2, where 0 indicates no filtering to be applied; 1 indicates weak filtering to be applied; and 2 indicates strong filtering to be applied. A bS value may be determined for a boundary having dimensions of 4x4 pixels, but mapped to a boundary having dimensions of 8x8 pixels. For an 8-pixel segment on a boundary in an 8x8 pixel grid, a bS value of the entire segment may be set to the larger of two bS values for two 4-pixel segments making up the 8-pixel segment.

[0051] Conventionally, the deblocking filter 212 may only determine a bS value of 2 in the case that block P or block Q is an intra-coded block.

[0052] According to an example embodiment of the present disclosure, the deblocking filter 212 may also determine a bS value of 2 in the case that block P or block Q is an inter-coded block rather than an intra-coded block, and has a resolution different from a resolution of the current frame. According to another example embodiment of the present disclosure, the deblocking filter 212 may determine a bS value of 1 in the case that block P or block Q is an inter-coded block rather than an intra-coded block, and has a resolution different from a resolution of the current frame. FIGS. 4A and 4B illustrate flowcharts of deblocking filter logic according to example embodiments of the present disclosure.

[0053] Further examples of deblocking filter logic according to example embodiments of the present disclosure with reference to Tables 4, 5, 6, 7, 8, 9, and 10. For the purpose of illustration, and without limitation to the context thereof, these example embodiments are based on deblocking filter logic according to the VVC specification.

[0054] Based on the VVC specification, the deblocking filter 212 receives at least the following inputs: with reference to an upper-left sample of a current frame, a coordinate (xCb, yCb) locating an upper-left sample pixel of a current coding block relative to an upper-left sample of a current frame; a variable nCbW specifying width of the current coding block; a variable nCbH specifying height of the current coding block; a variable edgeType specifying whether a vertical edge (denoted by, for example, the value EDGE_VER) or a horizontal edge (denoted by, for example, the value EDGE_HOR) is filtered; a variable cIdx specifying the color component of the current coding block; and a two-dimensional array edgeFlags having dimensions (nCbW)x(nCbH).

[0055] The deblocking filter 212 may then determine bS values $bS[xD_i][yD_j]$ for xD_i coordinates ranging over $i = 0 \ldots xN$ and yD_j coordinates ranging over $j = yN$, where xD_i, yD_j, xN, and yN are determined as follows:

[0056] The deblocking filter 212 sets a variable gridSize:

$$gridSize = cIdx == 0 \,?\, 4 : 8$$

[0057] In the event that edgeType has value EDGE_VER:

$$xD_i = (i * gridSize)$$

$$yD_j = cIdx == 0 \ ? \ (j << 2) \ : \ (j << 1)$$

$$xN \text{ is set equal to } Max(0, (nCbW / gridSize) - 1)$$

$$yN = cIdx == 0 \ ? \ (nCbH / 4) - 1 \ : \ (nCbH / 2) - 1$$

[0058] Otherwise, edgeType has value EDGE_HOR:

$$xD_i = cIdx == 0 \ ? \ (i << 2) \ : \ (i << 1)$$

$$yD_j = (j * gridSize)$$

$$xN = cIdx == 0 \ ? \ (nCbW / 4) - 1 \ : \ (nCbW / 2) - 1$$

$$yN = Max(0, (nCbH / gridSize) - 1)$$

[0059] Thus, for xD_i coordinates ranging over i = 0 . . . xN and yD_j coordinates ranging over j = yN, bS values $bS[xD_i][yD_j]$ are set as follows:

[0060] For xD_i and yD_j where $edgeFlags[xD_i][yD_j]$ is 0, $bS[xD_i][yD_j]$ is 0.

[0061] For xD_i and yD_j where $edgeFlags[xD_i][yD_j]$ is not 0, first sample values p_0 and q_0 are derived. In the event that edgeType has value EDGE_VER, p_0 is set to $recPicture[xCb + xD_i - 1][yCb + yD_j]$ and q_0 is set to $recPicture[xCb + xD_i][yCb + yD_j]$. Otherwise, edgeType has value EDGE_HOR, and p_0 is set to $recPicture[xCb + xD_i][yCb + yD_j - 1]$ and q_0 is set to $recPicture[xCb + xD_i][yCb + yD_j]$. Next, bS values $bS[xD_i][yD_j]$ are set as follows:

[0062] If cIdx has value 0 and both samples p_0 and q_0 are in respective coding blocks with intra_bdpcm_luma_flag having value 1, $bS[xD_i][yD_j]$ is set to 0.

[0063] Otherwise, in the event that cIdx has value greater than 0 and both sampled pixels p_0 and q_0 are in respective coding blocks with intra_bdpcm_chroma_flag having value 1, $bS[xD_i][yD_j]$ is set to 0.

[0064] Otherwise, in the event that a sampled pixel p_0 or q_0 is in a coding block of a coding unit coded by intra prediction mode, $bS[xD_i][yD_j]$ is set to 2.

[0065] Otherwise, in the event that the block edge is also a transform block edge and a sampled pixel p_0 or q_0 is in a coding block with ciip_flag having value 1, $bS[xD_i][yD_j]$ is set to 2.

[0066] Otherwise, in the event that the block edge is also a transform block edge and the sampled pixel p_0 or q_0 is in a transform block which contains one or more non-zero transform coefficient levels, $bS[xD_i][yD_j]$ is set equal to 1.

[0067] Otherwise, in the event that a prediction mode of a first coding sub-block containing the sampled pixel p_0 is different from a prediction mode of a second coding sub-block containing the sampled pixel q_0 (i.e., one of these coding sub-blocks is coded in IBC prediction mode and the other of the coding sub-blocks is coded in inter prediction mode), $bS[xD_i][yD_j]$ is set to 1.

[0068] Otherwise, in the event that cIdx has value 0, $edgeFlags[xD_i][yD_j]$ has value 2, and one or more of the following conditions are true, $bS[xD_i][yD_j]$ is set to 1:

[0069] A first coding sub-block containing the sampled pixel p_0 and a second coding sub-block containing the sampled pixel q_0 are both coded by IBC prediction mode, and an absolute difference between the horizontal or vertical component of the block vectors used in motion prediction of the two coding sub-blocks is greater than or equal to 8 in units of 1/16 luma sampled pixels; and/or:

[0070] In motion prediction of a first coding sub-block containing the sampled pixel p_0, different reference pictures or a different number of motion vectors are used than in motion prediction of a second coding sub-block containing the sampled pixel q_0; and/or:

[0071] (Herein, determination of whether the reference pictures used for the two coding sub-blocks are same or different may be based only on which pictures are referenced, without regard to whether a prediction is formed using an index into reference picture list 0 or an index into reference picture list 1, and also without regard to whether the index position within a reference picture list is different; and

[0072] Herein, the number of motion vectors that are used in motion prediction of a coding sub-block with upper-left sample covering (xSb, ySb), is equal to PredFlagL0[xSb][ySb] + PredFlagL1[xSb][ySb].)

[0073] A first motion vector is used in motion prediction of a first coding sub-block containing the sample p_0 and a second motion vector is used in motion prediction of a second coding sub-block containing the sample q_0, and an absolute difference between the horizontal component or vertical component of the first and the second motion vectors is greater than or equal to 8 in units of 1/16 luma sampled pixels; and/or:

[0074] A first and a second motion vector and a first and a second reference picture are used in motion prediction of a first coding sub-block containing the sample p_0, a third and a fourth motion vector for the first and the second reference pictures are used in motion prediction of a second coding sub-block containing the sample q_0, and an absolute difference between the horizontal or vertical component of two respective motion vectors used in motion prediction of the two coding sub-blocks for either

reference picture is greater than or equal to 8 in units of 1/16 luma sampled pixels; and/or:

[0075] A first and a second motion vector for a first reference picture are used in motion prediction of a first coding sub-block containing the sampled pixel p_0, a third and a fourth motion vector for a second reference picture are used in motion prediction of a second coding sub-block containing the sampled pixel q_0, and both following conditions are true:

[0076] An absolute difference between the horizontal or vertical component of a list 0 motion vector used in motion prediction of the two coding sub-blocks is greater than or equal to 8 in 1/16 luma sampled pixels, or an absolute difference between the horizontal or vertical component of a list 1 motion vector used in motion prediction of the two coding sub-blocks is greater than or equal to 8 in units of 1/16 luma sampled pixels; and:

[0077] An absolute difference between the horizontal or vertical component of a list 0 motion vector used in motion prediction of a first coding sub-block containing the sample p_0 and a list 1 motion vector used in motion prediction of a second coding sub-block containing the sample q_0 is greater than or equal to 8 in units of 1/16 luma sampled pixels, or an absolute difference between the horizontal or vertical component of a list 1 motion vector used in motion prediction of the first coding sub-block containing the sample p_0 and a list 0 motion vector used in motion prediction of the second coding sub-block containing the sample q_0 is greater than or equal to 8 in units of 1/16 luma sampled pixels.

[0078] In the event that none of the above conditions applies, $bS[xD_i][yD_j]$ is set to 0.

[0079] Furthermore, as illustrated by FIG. 5, within every two pairs of P and Q blocks across a CTU boundary, the deblocking filter 212 may determine a bS value of a block by referencing blocks left and above the block within the two pairs of blocks, thus reducing memory requirements for computation.

[0080] bS values determined in step 304 may subsequently be referenced by the deblocking filter 212 in step 312B to determine whether the deblocking filter 212 should apply strong filtering, or not.

[0081] According to example embodiments of the present disclosure, further modifications are made to the above process of setting values of $bS[xD_i][yD_j]$:

[0082] According to an example embodiment of the present disclosure, in the event that $edgeFlags[xD_i][yD_j]$ has value 2, and a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sampled pixel p_0 or q_0 is different from a resolution of the current picture, $bS[xD_i][yD_j]$ is set to 2, and Table 3 subsequently described with reference to step 312B is further modified as shown by Table 4.

[0083] According to another example embodiment of the present disclosure, the one or more of the following conditions wherein, if true, $bS[xD_i][yD_j]$ is set to 1 further includes: a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sample p_0 or q_0 being different than a resolution of the current picture, and Table 3 subsequently described with reference to step 312B is further modified as shown by Table 5.

[0084] According to another example embodiment of the present disclosure, in the event that $edgeFlags[xD_i][yD_j]$ has value 2, and a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sampled pixel p_0 is different from a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sampled pixel q_0, $bS[xD_i][yD_j]$ is set to 2, and Table 3 subsequently described with reference to step 312B is further modified as shown by Table 6.

[0085] According to another example embodiment of the present disclosure, in the event that $edgeFlags[xD_i][yD_j]$ has value 2, and in the event that at least one of the following conditions is true, $bS[xD_i][yD_j]$ is set to 2, and Table 3 subsequently described with reference to step 312B is further modified as shown by Table 7: a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sampled pixel p_0 or q_0 is lower than a resolution of the current

picture, or; a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sample p_0 is different from a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sampled pixel q_0.

[0086] According to another example embodiment of the present disclosure, the one or more of the following conditions wherein, if true, $bS[xD_i][yD_j]$ is set to 1 further includes: a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sample p_0 or q_0 being lower than a resolution of the current picture, and Table 3 subsequently described with reference to step 312B is further modified as shown by Table 8.

[0087] According to another example embodiment of the present disclosure, in the event that $edgeFlags[xD_i][yD_j]$ has value 2, and in the event that at least one of the following conditions is true, $bS[xD_i][yD_j]$ is set to 2, and Table 3 subsequently described with reference to step 312B is further modified as shown by Table 9: a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sampled pixel p_0 or q_0 is higher than a resolution of the current picture, or; a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sample p_0 is different from a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sampled pixel q_0.

[0088] According to another example embodiment of the present disclosure, the one or more of the following conditions wherein, if true, $bS[xD_i][yD_j]$ is set to 1 further includes: a resolution of one of the reference pictures used in motion prediction of a coding sub-block containing the sample p_0 or q_0 being higher than a resolution of the current picture, and Table 3 subsequently described with reference to step 312B is further modified as shown by Table 10.

[0089] With regard to the example embodiments as described above, generally, those example embodiments wherein $bS[xD_i][yD_j]$ is set to 2 may describe conditions wherein blocking artifacts are expected to be severe as a result of resolution differences, and thus the deblocking filter 212 should apply a strong filter. Those example

embodiments wherein $bS[xD_i][yD_j]$ is set to 1 may describe conditions wherein blocking artifacts are expected to be moderate as a result of resolution differences, and thus the deblocking filter 212 should not apply a strong filter.

[0090] At step 306A, performed according to the HEVC specification, the deblocking filter 212 determines threshold values β and t_C. The threshold values β and t_C may be utilized in the subsequent steps 308, 310, and 312 to control strength of the deblocking filter 212. The threshold values β and t_C may be determined by lookup of corresponding values β' and t_C' from a table such as Table 1 below according to the HEVC specification, based on a value of a luma quantization parameter Q (also referred to as qP_L).

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
β'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7	8
t_C'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Q	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
β'	9	10	11	12	13	14	15	16	17	18	29	22	24	26	28	30	32	34	36
t_C'	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4
Q	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53			
β'	38	40	42	44	46	48	50	52	54	56	58	60	62	64	-	-			
t_C'	5	5	6	6	7	8	9	10	11	13	14	16	18	20	22	24			

[0091] Values of t_C may be determined from values of t_C' by the following equation:

$$t_C = BitDepth < 10\ ?\ (t_C' + 2) \gg (10 - BitDepth)$$
$$:\ t_C' * \left(1 \ll (BitDepth - 10)\right)$$

[0092] According to the VVC specification, Table 1 may be modified by extending Q values through to a maximum of 63, by extending β' values as follows, and by

replacing t_C' values with the following (corresponding to Q values 0 through 63, in order):

[0093] [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 4, 4, 4, 4, 5, 5, 5, 5, 7, 7, 8, 9, 10, 10, 11, 13, 14, 15, 17, 19, 21, 24, 25, 29, 33, 36, 41, 45, 51, 57, 64, 71, 80, 89, 100, 112, 125, 141, 157, 177, 198, 222, 250, 280, 314, 352, 395]

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
β′	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
t_C′	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
β′	7	8	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28
t_C′	0	3	4	4	4	4	5	5	5	5	7	7	8	9	10	10	11
Q	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
β′	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
t_C′	13	14	15	17	19	21	24	25	29	33	36	41	45	51	57	64	71
Q	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65		
β′	64	66	68	70	72	74	76	78	80	82	84	86	88	-	-		
t_C′	80	89	100	112	125	141	157	177	198	222	250	280	314	352	395		

[0094] Q may be determined based on pixel samples from reconstructed luma sub-blocks of the current CU.

$$Q = Clip3\big(0, 65, qP + 2 * (bS - 1) + (slice\_tc\_offset\_div2 \ll 1)\big)$$

[0095] A value of β may be derived from β' as follows:

$$\beta = \beta' * \big(1 \ll (BitDepth_Y - 8)\big)$$

[0096] A value of t_C may be derived from t_C' as follows:

$$t_C = BitDepth < 10\ ?\ (t_C' + 2) \gg (10 - BitDepth) :\ t_C' * \big(1 \ll (BitDepth_Y - 10)\big)$$

[0097] At step 306B, performed according to the VVC specification, regardless of the deblocking filter 212 determining threshold values β and t_C, the deblocking filter 212 applies an offset to the luma quantization parameter qP_L. According to the VVC

specification, this offset may take precedence in controlling strength of the deblocking filter 212 over the effects of β and t_C.

[0098] An offset *qpOffset* may be derived from luma levels ("*LL*") of pixel samples from luma sub-blocks of the current CU as follows:

$$LL = \left(\left(p_{0,0} + p_{0,3} + q_{0,0} + q_{0,3} \right) \gg 2 \right) / (1 \ll bitDepth)$$

[0099] FIGS. 6A and 6B, described subsequently, illustrate the coordinates of those particular *p* and *q* pixels from which luma levels are sampled from.

[00100] From the value of LL as derived above, a transfer function may be applied to derive the offset *qpOffset*. A base value of *qpOffset* may be derived from a flag sps_ladf_lowest_interval_qp_offset in a slice header of the frame. The slice header further contains flags specifying lower bounds of multiple luma intensity level intervals. For each of these intervals, if LL exceeds the lower bound set for the interval, the base value of *qpOffset* is offset by an offset value in the range of -63 to 63, inclusive, preset for the interval in an offset array recorded in the slice header.

[00101] According to the VVC specification, qP_L may be derived as follows:

$$qP_L = \left(\left(Qp_Q + Qp_P + 1 \right) \gg 1 \right)$$

[00102] Where Qp_Q is a quantization parameter of a coding block containing the pixel $q_{0,0}$ and Qp_P is a quantization parameter of a coding block containing the pixel $p_{0,0}$, as FIGS. 6A and 6B illustrate subsequently.

[00103] According to example embodiments of the present disclosure, qP_L may be derived as follows:

$$qP_L = \left(\left(Qp_Q + Qp_P + 1 \right) \gg 1 \right) + qpOffset$$

[00104] At step 308A, the deblocking filter 212 determines whether the deblocking filter 212 is active for a first four vertical lines of pixels or horizontal lines of pixels running across an 8x8 boundary, and whether the deblocking filter is active for a second four vertical lines of pixels or horizontal lines of pixels running across the 8x8 boundary.

[00105] At step 308B, the deblocking filter 212 determines whether the deblocking filter 212 is active for a first four vertical lines of pixels or horizontal lines of pixels running across an 8x8 or 4x4 boundary, and whether the deblocking filter is active for a second four vertical lines of pixels or horizontal lines of pixels running across the 8x8 or 4x4 boundary.

[00106] FIGS. 6A and 6B illustrate determining whether the deblocking filter 212 is active for a first four horizontal lines and a second four horizontal lines running across a vertical 8x8 boundary according to example embodiments of the present disclosure, the lines being numbered from 0 to 7; for vertical lines, whether the deblocking filter 212 is active may be derived similarly. For each line, six pixels of the line on either side of the boundary are sampled. As illustrated by FIGS. 6A and 6B, the deblocking filter 212 samples the pixels $p2_0$, $p1_0$, $p0_0$, $q0_0$, $q1_0$, and $q2_0$ of the first line and the pixels $p2_3$, $p1_3$, $p0_3$, $q0_3$, $q1_3$, and $q2_3$ in the fourth line among the first four lines to determine whether the deblocking filter is active for the first four lines.

[00107] From the intensity values of these pixels, the following calculations are performed with regard to the first four lines.

$$dp0 = |p2_0 - 2 * p1_0 + p0_0|$$
$$dp3 = |p2_3 - 2 * p1_3 + p0_3|$$
$$dq0 = |q2_0 - 2 * q1_0 + q0_0|$$
$$dq3 = |q2_3 - 2 * q1_3 + q0_3|$$

[00108] In the case the sum of dp0, dp3, dq0, and dq3 is less than the value of β, the deblocking filter 212 will be active for the first four lines, and furthermore, the following variables are also set as inputs for filters.

[00109] The variable dE is set equal to 1.

[00110] If $dp0 + dp3 < (\beta + (\beta \gg 1)) \gg 3$, the variable dEp1 is set equal to 1.

[00111] If $dq0 + dq3 < (\beta + (\beta \gg 1)) \gg 3$, the variable dEq1 is set equal to 1.

[00112] In the case the sum of dp0, dp3, dq0, and dq3 is not less than the value of β, the deblocking filter 212 will not be active for the first four lines.

[00113] The deblocking filter 212 also samples the pixels $p2_4$, $p1_4$, $p0_4$, $q0_4$, $q1_4$, and $q2_4$ in the first line and the pixels $p2_7$, $p1_7$, $p0_7$, $q0_7$, $q1_7$, and $q2_7$ in the fourth line among

the second four lines to determine whether the deblocking filter is active for the second four horizontal lines. This is performed in a manner similar to the above-mentioned process for the first four lines.

[00114] At step 310A, performed according to the HEVC specification, the deblocking filter 212 determines whether strong or weak filtering is applied for the first four vertical lines or horizontal lines in the 8x8 boundary in the case that the deblocking filter 212 is active for those lines, and whether strong or weak filtering is applied for the second four vertical lines or horizontal lines in the 8x8 boundary in the case that the deblocking filter 212 is active for those lines.

[00115] At step 310B, performed according to the VVC specification, the deblocking filter 212 determines whether strong or weak filtering is applied for the first four vertical lines or horizontal lines in the 8x8 or 4x4 boundary in the case that the deblocking filter 212 is active for those lines, and whether strong or weak filtering is applied for the second four vertical lines or horizontal lines in the 8x8 or 4x4 boundary in the case that the deblocking filter 212 is active for those lines.

[00116] The deblocking filter 212 applies strong filtering to the first four lines if the following two sets of conditions are met, and weak filtering otherwise.

$$2 * (dp0 + dq0) < (\beta \gg 2), |p3_0 - p0_0| + |q0_0 - q3_0| < (\beta \gg 3)\ and\ |p0_0 - q0_0| < (5 * t_C + 1) \gg 1$$

$$2 * (dp3 + dq3) < (\beta \gg 2), |p3_3 - p0_3| + |q0_3 - q3_3| < (\beta \gg 3)\ and\ |p0_3 - q0_3| < (5 * t_C + 1) \gg 1$$

[00117] The deblocking filter 212 determines whether to apply strong or weak filtering to the second four lines in a manner similar to the above-mentioned process for the first four lines.

[00118] At step 312A, performed according to the HEVC specification, the deblocking filter 212 applies a strong filter to vertical lines or horizontal lines wherein the deblocking filter 212 determined to apply a strong filter.

[00119] The strong filter is applied to three pixels p_0, p_1, and p_2 of the block P side of the boundary with four pixels total as input, outputting pixels p_0', p_1', and p_2', respectively; and three pixels q_0, q_1, and q_2 of the block Q side of the boundary with

four pixels total as input, outputting pixels q_0', q_1', and q_2', respectively. The outputs are derived as below.

$$p_0' = (p_2 + 2 * p_1 + 2 * p_0 + 2 * q_0 + q_1 + 4) \gg 3$$
$$q_0' = (p_1 + 2 * p_0 + 2 * q_0 + 2 * q_1 + q_2 + 4) \gg 3$$
$$p_1' = (p_2 + p_1 + p_0 + q_0 + 2) \gg 2$$
$$q_1' = (p_0 + q_0 + q_1 + q_2 + 2) \gg 2$$
$$p_2' = (2 * p_3 + 3 * p_2 + p_1 + p_0 + q_0 + 4) \gg 3$$
$$q_2' = (p_0 + q_0 + q_1 + 3 * q_2 + 2 * q_3 + 4) \gg 3$$

[00120] At step 312B, performed according to the VVC specification, the deblocking filter 212 applies a strong filter to vertical lines or horizontal lines wherein the deblocking filter 212 determined to apply a strong filter.

[00121] In step 312B, in addition to the strong filter as described above with reference to step 312A, the deblocking filter 212 may apply a filter according to the VVC specification, for luma CTBs in particular, to a sub-block boundary 4x4 in dimensions rather than a sub-block boundary 8x8 in dimensions, as described above with reference to step 302. For such a filter, rather than being applied to three pixels each of the respective blocks to each side of the boundary, instead the filter may be applied to one pixel each of the respective blocks to each side of the boundary, where a block to one side of the boundary has a width of 4 pixels or less in the event that the boundary is vertical, or a block to one side of the boundary has a height of 4 pixels or less in the event that the boundary is horizontal. Such implementations may handle blocking artifacts from rectangular transform shapes, and may facilitate concurrent computation of the deblocking filter 212 by parallel computing threads.

[00122] Additionally, in step 312B, the deblocking filter 212 may apply a stronger deblocking filter (for example, a bilinear filter) according to the VVC specification, for luma CTBs in particular, in the event that sampled pixels on either the P side or the Q side of the boundary belong to a large block and in the event that two further conditions are also satisfied. Large blocks may be those blocks where width of a horizontal edge is greater than or equal to 32 pixels, or those blocks where height of a vertical edge is greater than or equal to 32 pixels.

[00123] The two further conditions are determined as follows:

Condition2 = (d < β) ? TRUE: FALSE

Condition3 = StrongFilterCondition = (dpq is less than (β >> 2), $sp_3 + sq_3$ is less than (3*β >> 5), and Abs($p_0 - q_0$) is less than (5 * t_C + 1) >> 1) ? TRUE : FALSE

[00124] The outputs are then derived as follows, where values p_i are block boundary samples for i = 0 to i = Sp – 1 on either the P side or the Q side of the boundary, values q_j are block boundary samples for j = 0 to Sq – 1 likewise on either the P side or the Q side of the boundary, and p_i' and q_j' are outputs for those respective inputs:

$$p_i' = (f_i * Middle_{s,t} + (64 - f_i) * P_s + 32) \gg 6), clipped\ to\ p_i \pm tcPD_i \quad (3\text{-}1)$$

$$q_j' = (g_j * Middle_{s,t} + (64 - g_j) * Q_s + 32) \gg 6), clipped\ to\ q_j \pm tcPD_j \quad (3\text{-}2)$$

[00125] Wherein $tcPD_i$ and $tcPD_j$ are position-dependent clippings and g_j, f_i, $Middle_{s,t}$, P_s and Q_s are derived based on Table 2 below.

Sp, Sq 7, 7 (p side: 7, q side: 7)	$f_i = 59 - i * 9$, *can also be described as* $\boldsymbol{f} = \{59,50,41,32,23,14,5\}$ $g_j = 59 - j * 9$, *can also be described as* $\boldsymbol{g} = \{59,50,41,32,23,14,5\}$ $Middle_{7,7} = (2 * (p_o + q_o) + p_1 + q_1 + p_2 + q_2 + p_3 + q_3 + p_4 + q_4 + p_5 + q_5 + p_6 + q_6 + 8) \gg 4$ $P_7 = (p_6 + p_7 + 1) \gg 1, \quad Q_7 = (q_6 + q_7 + 1) \gg 1$
7, 3 (p side: 7 q side: 3)	$f_i = 59 - i * 9$, *can also be described as* $\boldsymbol{f} = \{59,50,41,32,23,14,5\}$ $g_j = 53 - j * 21$, *can also be described as* $\boldsymbol{g} = \{53,32,11\}$ $Middle_{7,3} = (2 * (p_o + q_o) + q_0 + 2 * (q_1 + q_2) + p_1 + q_1 + p_2 + p_3 + p_4 + p_5 + p_6 + 8) \gg 4$ $P_7 = (p_6 + p_7 + 1) \gg 1, \quad Q_3 = (q_2 + q_3 + 1) \gg 1$
3, 7 (p side: 3 q side: 7)	$g_j = 59 - j * 9$, *can also be described as* $\boldsymbol{g} = \{59,50,41,32,23,14,5\}$ $f_i = 53 - i * 21$, *can also be described as* $\boldsymbol{f} = \{53,32,11\}$ $Middle_{3.7} = (2 * (q_o + p_o) + p_0 + 2 * (p_1 + p_2) + q_1 + p_1 + q_2 + q_3 + q_4 + q_5 + q_6 + 8) \gg 4$ $Q_7 = (q_6 + q_7 + 1) \gg 1, \quad P_3 = (p_2 + p_3 + 1) \gg 1$
7, 5 (p side: 7 q side: 5)	$g_j = 58 - j * 13$, *can also be described as* $\boldsymbol{g} = \{58,45,32,19,6\}$ $f_i = 59 - i * 9$, *can also be described as* $\boldsymbol{f} = \{59,50,41,32,23,14,5\}$ $Middle7,5 = (2 * (p_o + q_o + p_1 + q_1) + q_2 + p_2 + q_3 + p_3 + q_4 + p_4 + q_5 + p_5 + 8) \gg 4$ $Q_5 = (q_4 + q_5 + 1) \gg 1, \quad P_7 = (p_6 + p_7 + 1) \gg 1$

5, 7 (p side: 5 q side: 7)	$g_j = 59 - j * 9$, *can also be described as* $\boldsymbol{g} = \{59,50,41,32,23,14,5\}$ $f_i = 58 - i * 13$, *can also be described as* $\boldsymbol{f} = \{58,45,32,19,6\}$ $Middle5,7 = (2 * (q_o + p_o + p_1 + q_1) + q_2 + p_2 + q_3 + p_3 + q_4 + p_4 + q_5 + p_5 + 8) \gg 4$ $Q_7 = (q_6 + q_7 + 1) \gg 1, \quad P_5 = (p_4 + p_5 + 1) \gg 1$
5, 5 (p side: 5 q side: 5)	$g_j = 58 - j * 13$, *can also be described as* $\boldsymbol{g} = \{58,45,32,19,6\}$ $f_i = 58 - i * 13$, *can also be described as* $\boldsymbol{f} = \{58,45,32,19,6\}$ $Middle5,5 = (2 * (q_o + p_o + p_1 + q_1 + q_2 + p_2) + q_3 + p_3 + q_4 + p_4 + 8) \gg 4$ $Q_5 = (q_4 + q_5 + 1) \gg 1, \quad P_5 = (p_4 + p_5 + 1) \gg 1$
5, 3 (p side: 5 q side: 3)	$g_j = 53 - j * 21$, *can also be described as* $\boldsymbol{g} = \{53,32,11\}$ $f_i = 58 - i * 13$, *can also be described as* $\boldsymbol{f} = \{58,45,32,19,6\}$ $Middle5,3 = (q_o + p_o + p_1 + q_1 + q_2 + p_2 + q_3 + p_3 + 4) \gg 3$ $Q_3 = (q_2 + q_3 + 1) \gg 1, \quad P_5 = (p_4 + p_5 + 1) \gg 1$
3, 5 (p side: 3 q side: 5)	$g_j = 58 - j * 13$, *can also be described as* $\boldsymbol{g} = \{58,45,32,19,6\}$ $f_i = 53 - i * 21$, *can also be described as* $\boldsymbol{f} = \{53,32,11\}$ $Middle3,5 = (q_o + p_o + p_1 + q_1 + q_2 + p_2 + q_3 + p_3 + 4) \gg 3$ $Q_5 = (q_4 + q_5 + 1) \gg 1, \quad P_3 = (p_2 + p_3 + 1) \gg 1$

[00126] Additionally, in step 312B, the deblocking filter 212 may apply a stronger deblocking filter according to the VVC specification, for chroma CTBs in particular, to a sub-block boundary 8x8 in dimensions as described above with reference to step 302, in the event that both the P side and the Q side of the chroma CTB boundary have dimensions greater than or equal to 8 pixels of chroma sample and in the event that three further conditions are also satisfied.

[00127] The first condition is satisfied by a determination to apply strong filtering as described below with reference to Table 3, and the deblocking filter 212 determining in step 312B as described above that sampled pixels on both the P side and the Q side of the chroma CTB boundary belong to large blocks.

[00128] Table 3 below describes a decision-making process wherein the deblocking filter 212 may determine to apply strong filtering, or may not. "Adjacent blocks" may refer to the block on the P side and the block on the Q side of the filtered boundary. Where any of the Y, U, or V bS values in the rightmost three columns is determined as 2, the first condition may be satisfied. Where any of the Yu, U, or V bS values in the rightmost three columns is determined as 1, and sampled pixels on both the P side and

the Q side of the chroma boundary are determined to belong to large blocks, the first condition may also be satisfied.

Priority	Conditions	Y	U	V
5	At least one of the adjacent blocks is intra	2	2	2
4	At least one of the adjacent blocks has non-zero transform coefficients	1	1	1
3	Absolute difference between the motion vectors that belong to the adjacent blocks is greater than or equal to one half luma sample	1	N/A	N/A
2	Motion prediction in the adjacent blocks refers to vectors is different	1	N/A	N/A
1	Otherwise	0	0	0

[00129] Table 4 below describes a decision-making process according to another example embodiment of the present disclosure, as referenced above.

Priority	Conditions	Y	U	V
6	At least one of the adjacent blocks is intra	2	2	2
5	Resolution of at least one reference picture used in motion prediction of the adjacent blocks is different from resolution of current picture	2	2	2

4	At least one of the adjacent blocks has non-zero transform coefficients	1	1	1
3	Absolute difference between the motion vectors that belong to the adjacent blocks is greater than or equal to one half luma sample	1	N/A	N/A
2	Motion prediction in the adjacent blocks refers to vectors is different	1	N/A	N/A
1	Otherwise	0	0	0

[00130] Table 5 below describes a decision-making process according to another example embodiment of the present disclosure, as referenced above.

Priority	Conditions	Y	U	V
6	At least one of the adjacent blocks is intra	2	2	2
5	At least one of the adjacent blocks has non-zero transform coefficients	1	1	1
4	Resolution of at least one reference picture used in motion prediction of the adjacent blocks is different from resolution of current picture	1	N/A	N/A
3	Absolute difference between the motion vectors that belong to the adjacent blocks is greater than or equal to one half luma sample	1	N/A	N/A

2	Motion prediction in the adjacent blocks refers to vectors is different	1	N/A	N/A
1	Otherwise	0	0	0

[00131] Table 6 below describes a decision-making process according to another example embodiment of the present disclosure, as referenced above.

Priority	Conditions	Y	U	V
6	At least one of the adjacent blocks is intra	2	2	2
5	Resolutions of the reference pictures referenced in motion prediction of the adjacent blocks are different	2	2	2
4	At least one of the adjacent blocks has non-zero transform coefficients	1	1	1
3	Absolute difference between the motion vectors that belong to the adjacent blocks is greater than or equal to one half luma sample	1	N/A	N/A
2	Motion prediction in the adjacent blocks refers to vectors is different	1	N/A	N/A
1	Otherwise	0	0	0

[00132] Table 7 below describes a decision-making process according to another example embodiment of the present disclosure, as referenced above.

Priority	Conditions	Y	U	V
6	At least one of the adjacent blocks is intra	2	2	2
5	Resolution of at least one reference picture used in motion prediction of the adjacent blocks is lower than resolution of current picture; or, resolutions of reference pictures referenced in motion prediction of the adjacent blocks are different	2	2	2
4	At least one of the adjacent blocks has non-zero transform coefficients	1	1	1
3	Absolute difference between the motion vectors that belong to the adjacent blocks is greater than or equal to one half luma sample	1	N/A	N/A
2	Motion prediction in the adjacent blocks refers to vectors is different	1	N/A	N/A
1	Otherwise	0	0	0

[00133] Table 8 below describes a decision-making process according to another example embodiment of the present disclosure, as referenced above.

Priority	Conditions	Y	U	V
6	At least one of the adjacent blocks is intra	2	2	2
5	At least one of the adjacent blocks has non-zero transform coefficients	1	1	1

4	Resolution of at least one reference picture used in motion prediction of the adjacent blocks is lower than resolution of current picture	1	N/A	N/A
3	Absolute difference between the motion vectors that belong to the adjacent blocks is greater than or equal to one half luma sample	1	N/A	N/A
2	Motion prediction in the adjacent blocks refers to vectors is different	1	N/A	N/A
1	Otherwise	0	0	0

[00134] Table 9 below describes a decision-making process according to another example embodiment of the present disclosure, as referenced above.

Priority	Conditions	Y	U	V
6	At least one of the adjacent blocks is intra	2	2	2
5	Resolution of at least one reference picture used in motion prediction of the adjacent blocks is higher than resolution of current picture; or, resolutions of reference pictures referenced in motion prediction of the adjacent blocks are different	2	2	2
4	At least one of the adjacent blocks has non-zero transform coefficients	1	1	1

3	Absolute difference between the motion vectors that belong to the adjacent blocks is greater than or equal to one half luma sample	1	N/A	N/A
2	Motion prediction in the adjacent blocks refers to vectors is different	1	N/A	N/A
1	Otherwise	0	0	0

[00135] Table 10 below describes a decision-making process according to another example embodiment of the present disclosure, as referenced above.

Priority	Conditions	Y	U	V
6	At least one of the adjacent blocks is intra	2	2	2
5	At least one of the adjacent blocks has non-zero transform coefficients	1	1	1
4	Resolution of at least one reference picture used in motion prediction of the adjacent blocks is higher than resolution of current picture	1	N/A	N/A
3	Absolute difference between the motion vectors that belong to the adjacent blocks is greater than or equal to one half luma sample	1	N/A	N/A
2	Motion prediction in the adjacent blocks refers to vectors is different	1	N/A	N/A
1	Otherwise	0	0	0

[00136] The second condition is satisfied by the deblocking filter 212 determining in step 308 as described above to be active across a boundary.

[00137] The third condition is satisfied by the deblocking filter 212 determining in step 310 as described above to apply strong filtering over the boundary.

[00138] At step 314, the deblocking filter 212 applies a weak filter to vertical lines or horizontal lines wherein the deblocking filter 212 determined to apply a weak filter.

[00139] To apply a weak filter, the deblocking filter 212 determines a value Δ.

$$\Delta = (9 * (q_0 - p_0) - 3 * (q_1 - p_1) + 8) \gg 4$$

[00140] Then, when the absolute value of Δ is less than t_C * 10, the weak filter is applied to pixels p_0 and q_0 on either side of the boundary, outputting pixels p_0' and q_0', respectively.

$$\Delta = Clip3(-t_C, t_C, \Delta)$$

$$p_0' = Clip1_Y(p_0 + \Delta)$$

$$q_0' = Clip1_Y(q_0 - \Delta)$$

[00141] Furthermore, depending on the previously calculated values of dEp1 and dEq1, the weak filter may be applied to either or both of pixels p_1 and q_1 on either side of the boundary, each with three pixels total as input, outputting either or both of pixels p_1' and q_1', respectively.

[00142] If dEp1 is equal to 1:

$$\Delta p = Clip3\left(-(t_C \gg 1), t_C \gg 1, \left(((p_2 + p_0 + 1) \gg 1) - p_1 + \Delta\right) \gg 1\right)$$

$$p_1' = Clip1_Y(p_1 + \Delta p)$$

[00143] If dEq1 is equal to 1:

$$\Delta q = Clip3\left(-(t_C \gg 1), t_C \gg 1, \left(((q_2 + q_0 + 1) \gg 1) - q_1 + \Delta\right) \gg 1\right)$$

$$q_1' = Clip1_Y(q_1 + \Delta q)$$

[00144] According to example embodiments of the present disclosure implementing VVC, the above-described method 300 may be largely performed in a similar manner, except that the filter strength of a deblocking filter 212 may be further dependent upon averaged luma level of pixel samples of the reconstructed frame; the t_C' lookup table may be further extended; and stronger deblocking filters may be applied for both the luma and chroma CTBs. Further details of these processes need not be described for understanding of example embodiments of the present disclosure, and shall not be reiterated herein.

[00145] Next, a SAO filter may filter a CTB on a per-pixel basis by applying an offset to each pixel based on determining a SAO type of each pixel.

[00146] FIG. 7 illustrates an example flowchart of a SAO filter method 700 according to example embodiments of the present disclosure.

[00147] At step 702, a SAO filter 213 receives a frame and decides to apply SAO to a CTB of the frame.

[00148] A frame may store a flag *sao_type_idx* in a slice header of the frame, the value thereof indicating whether SAO is to be applied to the CTB, and, if so, which type of SAO is to be applied. A *sao_type_idx* value of 0 may indicate that SAO is not to be applied to a CTB of the frame; a *sao_type_idx* value of 1 may indicate that an edge offset filter, as described below, is to be applied to a CTB of the frame; and a *sao_type_idx* value of 2 may indicate that a band offset filter, as described below, is to be applied to a CTB of the frame.

[00149] According to example embodiments of the present disclosure, a *sao_type_idx* value of 3 may indicate that both edge offset and band offset are to be applied to a CTB of the frame.

[00150] Furthermore, each applicable CTB may have further SAO parameters stored including *sao_merge_left_flag*, *sao_merge_up_flag*, SAO type, and four offsets. A *sao_merge_left_flag* value of 1 for a CTB may denote that the SAO filter 213 should apply SAO type and offsets of a CTB left of the current CTB to the current CTB. A *sao_merge_up_flag* value of 1 for a CTB may indicate that the SAO filter 213 should apply SAO type and offsets of the CTB above the current CTB to the current CTB.

[00151] At step 704, the SAO filter 213 classifies a CTB as one of several SAO types.

[00152] Table 11 below illustrates that each CTB of a frame may be classified as type 0, in which case no SAO will be applied to the CTB, or may be classified as types 1 through 5, where in each case a different SAO will be applied to the CTB. Furthermore, for types 1 through 5, pixels of the CTB will be categorized into one of multiple categories.

SAO type	Sample adaptive offset type to be used	Number of categories
0	None	0
1	1-D 0-degree pattern edge offset	4
2	1-D 90-degree pattern edge offset	4
3	1-D 135-degree pattern edge offset	4
4	1-D 45-degree pattern edge offset	4
5	Band offset	4

[00153] Types 1 through 4 of CTBs are identified by an angle of an edge pattern of pixels including the current pixel *p* and two neighbor pixels. FIGS. 8A through 8D illustrate possible edge patterns that include the current pixel *p* and two neighbor pixels. FIG. 8A illustrates an edge pattern made up of pixels including the current pixel *p* and two neighbor pixels at a 0-degree angle. FIG. 8B illustrates an edge pattern made up of pixels including the current pixel *p* and two neighbor pixels at a 90-degree angle. FIG. 8C illustrates an edge pattern made up of pixels including the current pixel *p* and two neighbor pixels at a 135-degree angle. FIG. 8D illustrates an edge pattern made up of pixels including the current pixel *p* and two neighbor pixels at a 45-degree angle.

[00154] At step 706, in the case that a CTB is classified as a type for applying edge offset, the SAO filter 213 classifies a pixel of a CTB according to edge properties.

[00155] Each pixel has an 8-bit intensity value ranging from 0 through 255. The current pixel p may be classified by a comparison of its intensity with the two neighbor pixels (in either order) in accordance with Table 12 below.

Pixel condition	Classification	Meaning
None of the below	0	Largely monotonic
p < both neighbor pixels	1	Local minimum
p < first neighbor pixel and p == second neighbor pixel	2	Edge
p > first neighbor pixel and p == second neighbor pixel	3	Edge
p > both neighbor pixels	4	Local Maximum

[00156] According to an example embodiment of the present disclosure, the current pixel p may be classified by a comparison of its intensity with the two neighbor pixels (in either order, which determine cases 1 through 5 below), as well as with neighbor pixels in general in two opposing directions (which determines case 0 below) in accordance with Table 13 below.

Pixel condition	Classification	Meaning
Difference sum of neighbor pixels in one direction differs greatly or much smaller than difference sum of neighbor pixels in other direction	0	Strong edge

None of the below	1	Largely monotonic
p < both neighbor pixels	2	Local minimum
p < first neighbor pixel and p == second neighbor pixel	3	Edge
p > first neighbor pixel and p == second neighbor pixel	4	Edge
p > both neighbor pixels	5	Local Maximum

[00157] At step 708, based on pixel classification, the SAO filter 213 applies an offset to the current pixel based on an offset value. The offset value of the current pixel may be determined based on the classification of the current pixel. Furthermore, by classifying a strong edge (also referred to as a real edge) based on significant differences in pixels in two opposing directions, the SAO filter 213 may determine pixels on the strong edge that are likely to be smoothed during up-sampling, and apply an offset value to compensate for this behavior.

[00158] At step 710, in the case that a CTB is classified as a type for applying band offset, the SAO filter 213 classifies a pixel of a CTB into a band.

[00159] A pixel index over the entire range of pixel intensity values may be established by reducing all 8-bit pixel intensity values to their five most significant bits, thus equalizing all pixel intensity values within each of 32 bands, each covering a same-sized segment of the original range of pixel intensity values. Thus, each pixel lies within one of these 32 bands based on its pixel intensity value. Furthermore, each set of four adjacent bands may be grouped together, with each group being identified by its starting position counting from low to high values over the 32 bands.

[00160] At step 712, the SAO filter 213 applies an offset to each band based on an offset value. The offset value may be determined by the intensity value of the band. The offset value may reduce distortion of the band.

[00161] Next, an ALF 214 may filter a frame per 4x4 pixel sub-block of a luma CTB and a chroma CTB of a frame.

[00162] FIG. 9A illustrates an example flowchart of an ALF method 900 according to example embodiments of the present disclosure.

[00163] At step 902, an ALF 214 receives a frame and decides to apply ALF to a luma CTB and/or a chroma CTB of the frame.

[00164] A luma CTB has a flag to indicate whether ALF should be applied to the luma CTB. A chroma CTB may have a flag to indicate whether ALF should be applied to the chroma CTB. The ALF 214 may decide to apply ALF based on values of these flags.

[00165] A frame may store ALF filter parameters in a slice header of the frame. ALF filter parameters may include 25 sets of luma filter coefficients, which may be accordingly applied to luma CTBs based on classification thereof. According to example embodiments of the present disclosure, ALF filter parameters may include more than 25 sets of luma filter coefficients to accommodate more types of classification, such as 35 sets of luma filter coefficients derived from a classification scheme as described below.

[00166] Filter coefficients may be mapped to the pixels that make up the shape of the filter. As illustrated by FIG. 9B, a chroma filter 912 may have a 5x5 pixel diamond shape, and a luma filter 914 may have a 7x7 pixel diamond shape, with each pixel showing an assigned filter coefficient value.

[00167] To reduce bit overhead, filter coefficients of different classifications may be merged to some extent. Filter coefficients may be quantized with norm equal to 128. To further reduce multiplication complexity, a bitstream conformance may be applied, wherein a coefficient value of a central position of a filter may fall within a range of 0 through 2^8, and coefficient values of all other positions of the filter may fall within a range of -2^7 to 2^7-1, inclusive.

[00168] At step 904, the ALF 214 calculates gradient values of a sub-block of the luma CTB in multiple directions by obtaining reconstructed samples.

[00169] Starting from an upper left pixel (i, j) of the sub-block in the frame, a 1-D Laplacian calculation may be performed in four different directions by obtaining reconstructed samples $R(x, y)$ at intervals from pixels (x, y) of the reconstructed frame. Based on 1-D Laplacian calculations, a horizontal gradient of the sub-block may be calculated as follows:

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|$$

[00170] A vertical gradient of the sub-block may be calculated as follows:

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|$$

[00171] A gradient of the sub-block in a first diagonal direction may be calculated as follows:

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D1_{k,l}, D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)|$$

[00172] A gradient of the sub-block in a second diagonal direction may be calculated as follows:

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|$$

[00173] Rather than sample over the entire 4x4 pixel sub-block, each of the above calculations may be performed as a subsampled 1-D Laplacian calculation, which is performed by subsampling over only the shaded portions of the sub-block as illustrated by FIGS. 9C with regard to a vertical direction, FIG. 9D with regard to a horizontal direction, and FIGS. 9E and 9F with regard to diagonal directions. The subsampled pixel positions may be in common for each of the four calculations.

[00174] Among the horizontal and vertical gradients, $g_{h,v}^{max}$ and $g_{h,v}^{min}$ are determined as the maximum and minimum values, respectively, among the horizontal and vertical

gradients g_h and g_v; $g_{d0,d1}^{max}$ and $g_{d0,d1}^{min}$ are determined as the maximum and minimum values, respectively, among the two diagonal gradients g_{d0} and g_{d1}.

[00175] At step 906, the ALF 214 classifies a sub-block of a luma CTB.

[00176] For each sub-block of a luma CTB, an ALF 214 classifies the sub-block into one of multiple classes based on a classification index C which is derived from a directionality D and a quantized value of activity $\hat{A}$ of the sub-block. The value of D represents a direction of local gradients in the sub-block, and the value of $\hat{A}$ represents activity of local gradients in the sub-block. C may be derived as follows.

$$C = 5D + \hat{A}$$

[00177] Sub-block of a chroma CTB are not classified.

[00178] From the four values $g_{h,v}^{max}$, $g_{h,v}^{min}$, $g_{d0,d1}^{max}$, and $g_{d0,d1}^{min}$, directionality D is set according to the following steps comparing the gradient values to each other and to two threshold values t_1 and t_2, providing D with a range of values from 0 through 4. When D has this range of possible values, 25 possible values may be derived for C from the above equation, corresponding to 25 different filters that may be applied to the sub-block.

[00179] 1. If $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{mi}$ and $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$, D is set to 0.

[00180] 2. If $g_{h,v}^{max}/g_{h,v}^{min} > g_{d0,d1}^{max}/g_{d0,d1}^{min}$, continue to step 3 below; otherwise continue to step 4 below.

[00181] 3. If $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$, D is set to 2; otherwise, D is set to 1.

[00182] 4. If $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$, D is set to 4; otherwise, D is set to 3.

[00183] According to example embodiments of the present disclosure, directionality D may be set according to the following steps instead comparing the gradient values to each other and to three threshold values t_1, t_2, and t_3, providing D with a range of values from 0 through 6. When D has a greater range of possible values than 0 to 4, more possible values than 25 may be derived for C from the above equation. For example, when D has a range of possible values from 0 to 6, 35 possible values may be derived for C from the above equation, corresponding to 35 different filters that may be applied to the sub-block.

[00184] 1. If $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ and $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$, D is set to 0.

[00185] 2. If $g_{h,v}^{max}/g_{h,v}^{min} > g_{d0,d1}^{max}/g_{d0,d1}^{min}$, continue to step 3 below; otherwise continue to step 5 below.

[00186] 3. If $g_{h,v}^{max} \leq t_2 \cdot g_{h,v}^{min}$, D is set to 1; otherwise continue to step 4 below.

[00187] 4. If $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ and $g_{h,v}^{max} \leq t_3 \cdot g_{h,v}^{min}$, D is set to 2; otherwise, D is set to 3.

[00188] 5. If $g_{d0,d1}^{max} \leq t_2 \cdot g_{d0,d1}^{min}$, D is set to 4; otherwise continue to step 6 below.

[00189] 6. If $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$ and $g_{d0,d1}^{max} \leq t_3 \cdot g_{d0,d1}^{min}$, D is set to 5; otherwise, D is set to 6.

[00190] According to other example embodiments of the present disclosure, directionality D may be set according to the following steps instead comparing the gradient values to each other, the maximum gradient among the gradient values, and to two threshold values t_1 and t_2, providing D with a range of values from 0 through 6.

[00191] 1. If $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ and $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$, D is set to 0.

[00192] 2. If $g_{h,v}^{max}/g_{h,v}^{min} > g_{d0,d1}^{max}/g_{d0,d1}^{min}$, continue to step 3 below; otherwise continue to step 6 below.

[00193] 3. If $g_{h,v}^{max} \leq t_2 \cdot g_{h,v}^{min}$ and the maximum gradient is horizontal, D is set to 1; otherwise continue to step 4 below.

[00194] 4. If $g_{h,v}^{max} \leq t_2 \cdot g_{h,v}^{min}$ and the maximum gradient is vertical, D is set to 2; otherwise continue to step 5 below.

[00195] 5. If $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ and the maximum gradient is horizontal, D is set to 3; otherwise, D is set to 4.

[00196] 6. If $g_{d0,d1}^{max} \leq t_2 \cdot g_{d0,d1}^{min}$, D is set to 5; otherwise, D is set to 6.

[00197] Furthermore, activity A is calculated by the following variation of the 1-D Laplacian calculation variation.

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} \left(V_{k,l} + H_{k,l}\right)$$

[00198] The value of activity A is quantized to a value $\hat{A}$ over a range of 0 through 4.

[00199] At step 908, prior to applying a filter to a sub-block, the ALF 214 applies one of several geometric transformations to filter coefficients of the filter. A geometric transformation may be chosen based on comparisons between gradient values according to Table 14 below.

Gradient value comparisons	Transformation
$g_{d2} < g_{d1}$, $g_h < g_v$	None
$g_{d2} < g_{d1}$, $g_v < g_h$	Diagonal
$g_{d1} < g_{d2}$, $g_h < g_v$	Vertical flip
$g_{d1} < g_{d2}$, $g_v < g_h$	Rotation

[00200] The above geometric transformations may be defined as the following functions:

[00201] Diagonal: $f_D(k,l) = f(l,k)$

[00202] Vertical flip: $f_V(k,l) = f(k, K-l-1)$

[00203] Rotation: $f_R(k,l) = f(K-l-1, k)$

[00204] K is the size of the filter, and $0 \leq k,l \leq K-1$ are coefficients coordinates, such that coordinate (0, 0) is at an upper left corner of the filter and coordinate (K–1, K–1) is at a lower right corner of the filter. Each transformation is applied to the filter coefficients $f(k, l)$ according to gradient values calculated as described above.

[00205] At step 910, the ALF 214 applies a filter having a filter coefficient $f(k, l)$ over each sub-block. For a luma sub-block, the filter coefficients to be applied may depend on the filter to be applied among all available filters, according to the classification index C. For chroma sub-blocks, the filter coefficients to be applied may be constant.

[00206] The filter may act upon a sample value $R(i, j)$ of a reconstructed frame, outputting a sample value $R'(i, j)$ as below.

$$R(i,j) = \left(\sum_{k=-L/2}^{L/2} \sum_{l=-L/2}^{L/2} f(k,l) \times R(i+k, j+l) + 64 \right) \gg 7$$

[00207] L is a filter length, $f_{m,n}$ denotes a filter coefficient, and $f(k, l)$ denotes a decoded filter coefficient.

[00208] FIG. 10 illustrates an example system 1000 for implementing the processes and methods described above for implementing resolution-adaptive video coding in deblocking filters.

[00209] The techniques and mechanisms described herein may be implemented by multiple instances of the system 1000 as well as by any other computing device, system, and/or environment. The system 1000 shown in FIG. 10 is only one example of a system and is not intended to suggest any limitation as to the scope of use or functionality of any computing device utilized to perform the processes and/or procedures described above. Other well-known computing devices, systems, environments and/or configurations that may be suitable for use with the embodiments include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, game consoles, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, implementations using field programmable gate arrays ("FPGAs") and application specific integrated circuits ("ASICs"), and/or the like.

[00210] The system 1000 may include one or more processors 1002 and system memory 1004 communicatively coupled to the processor(s) 1002. The processor(s) 1002 may execute one or more modules and/or processes to cause the processor(s) 1002 to perform a variety of functions. In some embodiments, the processor(s) 1002 may include a central processing unit (CPU), a graphics processing unit (GPU), both CPU and GPU, or other processing units or components known in the art. Additionally, each of the processor(s) 1002 may possess its own local memory, which also may store program modules, program data, and/or one or more operating systems.

[00211] Depending on the exact configuration and type of the system 1000, the system memory 1004 may be volatile, such as RAM, non-volatile, such as ROM, flash memory, miniature hard drive, memory card, and the like, or some combination thereof.

The system memory 1004 may include one or more computer-executable modules 1006 that are executable by the processor(s) 1002.

[00212] The modules 1006 may include, but are not limited to, a deblocking filter module 1008, which includes a boundary determining module 1010, a boundary strength determining module 1012, a threshold determining module 1014, an offset applying module 1016, a filter activity determining module 1018, a filter strength determining module 1020, a strong filter applying module 1022, and a weak filter applying module 1024.

[00213] The boundary determining module 1010 may be configured to determine block and sub-block boundaries to filter as abovementioned with reference to FIGS. 3A and 3B.

[00214] The boundary strength determining module 1012 may be configured to determines a bS value of a boundary being filtered as abovementioned with reference to FIGS. 3A and 3B.

[00215] The threshold determining module 1014 may be configured to determine threshold values, as abovementioned with reference to FIG. 3A.

[00216] The offset applying module 1016 may be configured to apply an offset to a luma quantization parameter, as abovementioned with reference to FIG. 3B.

[00217] The filter activity determining module 1018 may be configured to determine whether the deblocking filter module 1008 is active for a first four vertical lines of pixels or horizontal lines of pixels running across an 8x8 boundary, and whether the deblocking filter is active for a second four vertical lines of pixels or horizontal lines of pixels running across the 8x8 boundary, as abovementioned with reference to FIG. 3A, or may be configured to determine whether the deblocking filter module 1008 is active for a first four vertical lines of pixels or horizontal lines of pixels running across an 8x8 or 4x4 boundary, and whether the deblocking filter is active for a second four vertical lines of pixels or horizontal lines of pixels running across the 8x8 or 4x4 boundary, as abovementioned with reference to FIG. 3B.

[00218] The filter strength determining module 1020 may be configured to determine whether strong or weak filtering is applied for the first four vertical lines or horizontal

lines in the 8x8 boundary in the case that the deblocking filter module 1008 is active for those lines, and whether strong or weak filtering is applied for the second four vertical lines or horizontal lines in the 8x8 boundary in the case that the deblocking filter module 1008 is active for those lines, as abovementioned with reference to FIG. 3A, or may be configured to determine whether strong or weak filtering is applied for the first four vertical lines or horizontal lines in the 8x8 or 4x4 boundary in the case that the deblocking filter module 1008 is active for those lines, and whether strong or weak filtering is applied for the second four vertical lines or horizontal lines in the 8x8 or 4x4 boundary in the case that the deblocking filter module 1008 is active for those lines, as abovementioned with reference to FIG. 3B.

[00219] The strong filter applying module 1022 may be configured to apply a strong filter to vertical lines or horizontal lines wherein the deblocking filter module 1008 determined to apply a strong filter, as abovementioned with reference to FIGS. 3A and 3B.

[00220] The weak filter applying module 1024 may be configured to apply a weak filter to vertical lines or horizontal lines wherein the deblocking filter module 1008 determined to apply a weak filter, as abovementioned with reference to FIGS. 3A and 3B.

[00221] The system 1000 may additionally include an input/output (I/O) interface 1040 for receiving video source data and bitstream data, and for outputting reconstructed frames into a reference frame buffer, a transmission buffer, and/or a display buffer. The system 1000 may also include a communication module 1050 allowing the system 1000 to communicate with other devices (not shown) over a network (not shown). The network may include the Internet, wired media such as a wired network or direct-wired connections, and wireless media such as acoustic, radio frequency (RF), infrared, and other wireless media.

[00222] FIG. 11 illustrates an example system 1100 for implementing the processes and methods described above for implementing resolution-adaptive video coding in SAO filters.

[00223] The techniques and mechanisms described herein may be implemented by multiple instances of the system 1100 as well as by any other computing device, system, and/or environment. The system 1100 shown in FIG. 11 is only one example of a system and is not intended to suggest any limitation as to the scope of use or functionality of any computing device utilized to perform the processes and/or procedures described above. Other well-known computing devices, systems, environments and/or configurations that may be suitable for use with the embodiments include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, game consoles, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, implementations using field programmable gate arrays ("FPGAs") and application specific integrated circuits ("ASICs"), and/or the like.

[00224] The system 1100 may include one or more processors 1102 and system memory 1104 communicatively coupled to the processor(s) 1102. The processor(s) 1102 may execute one or more modules and/or processes to cause the processor(s) 1102 to perform a variety of functions. In some embodiments, the processor(s) 1102 may include a central processing unit (CPU), a graphics processing unit (GPU), both CPU and GPU, or other processing units or components known in the art. Additionally, each of the processor(s) 1102 may possess its own local memory, which also may store program modules, program data, and/or one or more operating systems.

[00225] Depending on the exact configuration and type of the system 1100, the system memory 1104 may be volatile, such as RAM, non-volatile, such as ROM, flash memory, miniature hard drive, memory card, and the like, or some combination thereof. The system memory 1104 may include one or more computer-executable modules 1106 that are executable by the processor(s) 1102.

[00226] The modules 1106 may include, but are not limited to, a SAO filter module 1108. The SAO filter module 1108 may include a filter application deciding module 1110, a CTB classifying module 1112, a pixel classifying module 1114, an edge offset

applying module 1116, a band classifying module 1118, and a band offset applying module 1120.

[00227] The filter application deciding module 1110 may be configured to receive a frame and decide to apply SAO to a CTB of the frame, as abovementioned with reference to FIG. 7.

[00228] The CTB classifying module 1112 may be configured to classify a CTB as one of several SAO types, as abovementioned with reference to FIG. 7.

[00229] The pixel classifying module 1114 may be configured to classify a pixel of a CTB according to edge properties in the case that a CTB is classified as a type for applying edge offset as abovementioned with reference to FIG. 7.

[00230] The edge offset applying module 1116 may be configured to apply an offset to the current pixel based on pixel classification and based on an offset value, as abovementioned with reference to FIG. 7.

[00231] The band classifying module 1118 may be configured to classify a pixel of a CTB into a band in the case that a CTB is classified as a type for applying band offset, as abovementioned with reference to FIG. 7.

[00232] The band offset applying module 1120 may be configured to apply an offset to each band based on an offset value, as abovementioned with reference to FIG. 7.

[00233] The system 1100 may additionally include an input/output (I/O) interface 1140 for receiving video source data and bitstream data, and for outputting reconstructed frames into a reference frame buffer, a transmission buffer, and/or a display buffer. The system 1100 may also include a communication module 1150 allowing the system 1100 to communicate with other devices (not shown) over a network (not shown). The network may include the Internet, wired media such as a wired network or direct-wired connections, and wireless media such as acoustic, radio frequency (RF), infrared, and other wireless media.

[00234] FIG. 12 illustrates an example system 1200 for implementing the processes and methods described above for implementing resolution-adaptive video coding in ALF.

[00235] The techniques and mechanisms described herein may be implemented by multiple instances of the system 1200 as well as by any other computing device, system, and/or environment. The system 1200 shown in FIG. 12 is only one example of a system and is not intended to suggest any limitation as to the scope of use or functionality of any computing device utilized to perform the processes and/or procedures described above. Other well-known computing devices, systems, environments and/or configurations that may be suitable for use with the embodiments include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, game consoles, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, implementations using field programmable gate arrays ("FPGAs") and application specific integrated circuits ("ASICs"), and/or the like.

[00236] The system 1200 may include one or more processors 1202 and system memory 1204 communicatively coupled to the processor(s) 1202. The processor(s) 1202 may execute one or more modules and/or processes to cause the processor(s) 1202 to perform a variety of functions. In some embodiments, the processor(s) 1202 may include a central processing unit (CPU), a graphics processing unit (GPU), both CPU and GPU, or other processing units or components known in the art. Additionally, each of the processor(s) 1202 may possess its own local memory, which also may store program modules, program data, and/or one or more operating systems.

[00237] Depending on the exact configuration and type of the system 1200, the system memory 1204 may be volatile, such as RAM, non-volatile, such as ROM, flash memory, miniature hard drive, memory card, and the like, or some combination thereof. The system memory 1204 may include one or more computer-executable modules 1206 that are executable by the processor(s) 1202.

[00238] The modules 1206 may include, but are not limited to, an ALF module 1208. The ALF module 1208 may include a filter application deciding module 1210, a gradient value calculating module 1212, a block classifying module 1214, a transformation applying module 1216, and a filter applying module 1218.

[00239] The filter application deciding module 1210 may be configured to receive a frame and decide to apply ALF to a luma CTB and/or a chroma CTB of the frame, as abovementioned with reference to FIG. 9.

[00240] The gradient value calculating module 1212 may be configured to calculate gradient values of the sub-block in multiple directions by obtaining reconstructed samples, as abovementioned with reference to FIG. 9.

[00241] The block classifying module 1214 may be configured to classify a sub-block of a luma CTB, as abovementioned with reference to FIG. 9.

[00242] The transformation applying module 1216 may be configured to apply one of several geometric transformations to filter coefficients of the filter, as abovementioned with reference to FIG. 9.

[00243] The filter applying module 1218 may be configured to apply a filter having a filter coefficient $f(k, l)$ over each sub-block, as abovementioned with reference to FIG. 9.

[00244] The system 1200 may additionally include an input/output (I/O) interface 1240 for receiving video source data and bitstream data, and for outputting reconstructed frames into a reference frame buffer, a transmission buffer, and/or a display buffer. The system 1200 may also include a communication module 1250 allowing the system 1200 to communicate with other devices (not shown) over a network (not shown). The network may include the Internet, wired media such as a wired network or direct-wired connections, and wireless media such as acoustic, radio frequency (RF), infrared, and other wireless media.

[00245] Some or all operations of the methods described above can be performed by execution of computer-readable instructions stored on a computer-readable storage medium, as defined below. The term "computer-readable instructions" as used in the description and claims, include routines, applications, application modules, program modules, programs, components, data structures, algorithms, and the like. Computer-readable instructions can be implemented on various system configurations, including single-processor or multiprocessor systems, minicomputers, mainframe computers,

personal computers, hand-held computing devices, microprocessor-based, programmable consumer electronics, combinations thereof, and the like.

[00246] The computer-readable storage media may include volatile memory (such as random-access memory (RAM)) and/or non-volatile memory (such as read-only memory (ROM), flash memory, etc.). The computer-readable storage media may also include additional removable storage and/or non-removable storage including, but not limited to, flash memory, magnetic storage, optical storage, and/or tape storage that may provide non-volatile storage of computer-readable instructions, data structures, program modules, and the like.

[00247] A non-transient computer-readable storage medium is an example of computer-readable media. Computer-readable media includes at least two types of computer-readable media, namely computer-readable storage media and communications media. Computer-readable storage media includes volatile and non-volatile, removable and non-removable media implemented in any process or technology for storage of information such as computer-readable instructions, data structures, program modules, or other data. Computer-readable storage media includes, but is not limited to, phase change memory (PRAM), static random-access memory (SRAM), dynamic random-access memory (DRAM), other types of random-access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technology, compact disk read-only memory (CD-ROM), digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other non-transmission medium that can be used to store information for access by a computing device. In contrast, communication media may embody computer-readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave, or other transmission mechanism. A computer-readable storage medium employed herein shall not be interpreted as a transitory signal itself, such as a radio wave or other free-propagating electromagnetic wave, electromagnetic waves propagating through a waveguide or other transmission

medium (such as light pulses through a fiber optic cable), or electrical signals propagating through a wire.

[00248] The computer-readable instructions stored on one or more non-transitory computer-readable storage media that, when executed by one or more processors, may perform operations described above with reference to FIGS. 1A-12. Generally, computer-readable instructions include routines, programs, objects, components, data structures, and the like that perform particular functions or implement particular abstract data types. The order in which the operations are described is not intended to be construed as a limitation, and any number of the described operations can be combined in any order and/or in parallel to implement the processes.

[00249] By the abovementioned technical solutions, the present disclosure provides inter-coded resolution-adaptive video coding supported by multiple in-loop filters, restoring high-frequency loss that occurs when a picture is down-sampled and subsequently up-sampled, and improving image quality during a resolution-adaptive video coding process. The methods and systems described herein provide a deblocking filter which takes resolution differences between frames undergoing motion prediction into account in determining filter strength, with further modifications for the next-generation video codec specification VVC. The deblocking filter may apply a strong filter or a weak filter in cases where a first reference frame referenced in motion prediction of a block adjacent to the block boundary has a resolution different from that of a second reference frame referenced in motion prediction of a block adjacent to the block boundary, that of a second reference frame referenced in motion prediction of the current frame, or that of the current frame.

EXAMPLE CLAUSES

[00250] A. A method comprising: receiving a current frame; determining a block boundary to be filtered within the current frame; determining a boundary strength of the block boundary based on a difference in resolution between a first reference frame referenced in motion prediction of a block adjacent to the block boundary and another frame; and applying a deblocking filter to the block boundary based on the boundary strength.

[00251] B. The method as paragraph A recites, wherein the second frame is a reference frame referenced in motion prediction of another block adjacent to the block boundary.

[00252] C. The method as paragraph A recites, wherein the second frame is a reference frame referenced in motion prediction of the current frame.

[00253] D. The method as paragraph A recites, wherein the second frame is the current frame.

[00254] E. The method as paragraph A recites, wherein the first reference frame has a different resolution than the second frame.

[00255] F. The method as paragraph A recites, wherein the first reference frame has a lower resolution than the second frame.

[00256] G. The method as paragraph A recites, wherein the first reference frame has a higher resolution than the second frame.

[00257] H. A method comprising: receiving a frame and deciding to apply SAO to a CTB of the frame; classifying the CTB as one of a plurality of SAO types; classifying a pixel of a CTB according to edge properties by at least comparing difference sums of neighbor pixels in two opposing directions; and applying an edge offset to the pixel based on an offset value.

[00258] I. The method as paragraph H recites, wherein deciding to apply SAO to a CTB of the frame comprises deciding to apply at least edge offset to the CTB based on a value of a flag stored in a slice header of the frame.

[00259] J. The method as paragraph I recites, wherein deciding to apply SAO to a CTB of the frame further comprises deciding to apply a band offset to the CTB based on the value of a flag stored in a slice header of the frame.

[00260] K. The method as paragraph J recites, further comprising classifying a pixel of a CTB into a band and applying an offset to the band based on an offset value.

[00261] L. The method as paragraph A recites, wherein the frame is received from an up-sampler.

[00262] M. A method comprising: receiving a frame and deciding to apply ALF to a CTB of the frame; calculating a plurality of gradient values of a block of the

CTB; determining a classification of the block based on computing a directionality value of at least six possible directionality values based on the plurality of gradient values; and applying a filter to the block, the filter comprising a set of filter coefficients determined by classification of the block.

[00263] N. The method as paragraph M recites, wherein the CTB is a luma CTB and the block is a luma block of the luma CTB.

[00264] O. The method as paragraph M recites, wherein the directionality value is computed by comparing the plurality of gradient values with at least three threshold values.

[00265] P. The method as paragraph M recites, wherein the directionality value is computed by comparing the plurality of gradient values with at least two threshold values and a maximum among the plurality of gradient values.

[00266] Q. The method as paragraph M recites, wherein a set of filter coefficients comprises a plurality of values arranged among 7x7 pixels.

[00267] R. The method as paragraph Q recites, wherein the set of filter coefficients is stored in a header of the frame.

[00268] S. The method as paragraph R recites, wherein the header stores more than 25 sets of filter coefficients and each set of filter coefficient corresponds to a classification of the block.

[00269] T. The method as paragraph M recites, wherein the frame is received from an up-sampler.

[00270] U. A system comprising: one or more processors; and memory communicatively coupled to the one or more processors, the memory storing computer-executable modules executable by the one or more processors that, when executed by the one or more processors, perform associated operations, the computer-executable modules including: a deblocking filter module configured to receive a current frame in a coding loop, the deblocking filter module further comprising a boundary determining module configured to determine a block boundary to be filtered within the current frame; a boundary strength determining module configured to determine a boundary strength of a block boundary to be filtered based on a difference in resolution between a first

reference frame referenced in motion prediction of a block adjacent to the block boundary and a second frame; and a strong filter applying module and a weak filter applying module each configured to apply a deblocking filter to the block boundary based on the boundary strength.

[00271] V. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a lower resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

[00272] W. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a lower resolution than the second frame, and the second frame being the current frame.

[00273] X. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a different resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of another block adjacent to the block boundary.

[00274] Y. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 1 based on the first reference frame having a lower resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

[00275] Z. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 1 based on the first reference frame having a lower resolution than the second frame, and the second frame being the current frame.

[00276] AA. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a higher resolution than the second

frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

[00277] BB. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a higher resolution than the second frame, and the second frame being the current frame.

[00278] CC. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a different resolution than the second frame, and the second frame being the current frame.

[00279] DD. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 1 based on the first reference frame having a higher resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

[00280] EE. The system as paragraph U recites, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 1 based on the first reference frame having a higher resolution than the second frame, and the second frame being the current frame.

[00281] BB. A system comprising: a SAO filter module configured to receive a frame, the SAO filter module further comprising a filter application deciding module configured to decide to apply SAO to a CTB of the frame; a CTB classifying module configured to classify the CTB as one of a plurality of SAO types; a pixel classifying module configured to classify a pixel of a CTB according to edge properties by at least comparing difference sums of neighbor pixels in two opposing directions; and an edge offset applying module configured to apply an edge offset to the pixel based on an offset value.

[00282] CC. The system as paragraph BB recites, wherein the filter application deciding module is further configured to decide to apply at least edge offset to the CTB based on a value of a flag stored in a slice header of the frame.

[00283] DD. The system as paragraph CC recites, wherein the filter application deciding module is further configured to decide to apply a band offset to the CTB based on the value of a flag stored in a slice header of the frame.

[00284] EE. The system as paragraph DD recites, further comprising a band classifying module configured to classify a pixel of a CTB into a band and a band offset applying module configured to apply an offset to the band based on an offset value.

[00285] FF. The system as paragraph EE recites, wherein the SAO filter module is configured to receive the frame from an up-sampler.

[00286] GG. A system comprising: an ALF module configured to receive a frame, the ALF module further comprising a filter application deciding module configured to decide to apply ALF to a CTB of the frame; a gradient value calculating module configured to calculate a plurality of gradient values of a block of the CTB; a block classifying module configured to determine a classification of the block based on computing a directionality value of at least six possible directionality values based on the plurality of gradient values; and a filter applying module configured to apply a filter to the block, the filter comprising a set of filter coefficients determined by classification of the block.

[00287] HH. The system as paragraph GG recites, wherein the CTB is a luma CTB and the block is a luma block of the luma CTB.

[00288] II. The system as paragraph GG recites, wherein the block classifying module is configured to compute a directionality value by comparing the plurality of gradient values with at least three threshold values.

[00289] JJ. The system as paragraph GG recites, wherein the block classifying module is configured to compute a directionality value by comparing the plurality of gradient values with at least two threshold values and a maximum among the plurality of gradient values.

[00290] KK. The system as paragraph GG recites, wherein a set of filter coefficients comprises a plurality of values arranged among 7x7 pixels.

[00291] LL. The system as paragraph KK recites, wherein the set of filter coefficients is stored in a header of the frame.

[00292] MM. The system as paragraph MM recites, wherein the header stores more than 25 sets of filter coefficients and each set of filter coefficient corresponds to a classification of the block.

[00293] NN. The system as paragraph GG recites, wherein the ALF module is configured to receive the frame from an up-sampler.

[00294] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the claims.

CLAIMS

WHAT IS CLAIMED IS:

1. A method comprising:

receiving a current frame;

determining block boundaries to be filtered within the current frame;

determining a boundary strength of a block boundary to be filtered based on a difference in resolution between a first reference frame referenced in motion prediction of a block adjacent to the block boundary and a second frame; and

applying a deblocking filter to the block boundary based on the boundary strength.

2. The method of claim 1, wherein the second frame is a reference frame referenced in motion prediction of another block adjacent to the block boundary.

3. The method of claim 1, wherein the second frame is a reference frame referenced in motion prediction of the current frame.

4. The method of claim 1, wherein the second frame is the current frame.

5. The method of claim 1, wherein the first reference frame has a different resolution than the second frame.

6. The method of claim 1, wherein the first reference frame has a lower resolution than the second frame.

7. The method of claim 1, wherein the first reference frame has a higher resolution than the second frame.

8. A system comprising:

one or more processors; and

memory communicatively coupled to the one or more processors, the memory storing computer-executable modules executable by the one or more processors that, when executed by the one or more processors, perform associated operations, the computer-executable modules including:

a deblocking filter module configured to receive a current frame in a coding loop, the deblocking filter module further comprising a boundary determining module configured to determine a block boundary to be filtered within the current frame;

a boundary strength determining module configured to determine a boundary strength of a block boundary to be filtered based on a difference in resolution between a first reference frame referenced in motion prediction of a block adjacent to the block boundary and a second frame; and

a strong filter applying module and a weak filter applying module each configured to apply a deblocking filter to the block boundary based on the boundary strength.

9. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a lower resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

10. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a lower resolution than the second frame, and the second frame being the current frame.

11. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a different resolution than the second frame, and the

second frame being a reference frame referenced in motion prediction of another block adjacent to the block boundary.

12. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 1 based on the first reference frame having a lower resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

13. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 1 based on the first reference frame having a lower resolution than the second frame, and the second frame being the current frame.

14. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a higher resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

15. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a higher resolution than the second frame, and the second frame being the current frame.

16. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 2 based on the first reference frame having a different resolution than the second frame, and the second frame being the current frame.

17. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 1 based on the first reference frame having a higher resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

18. The system of claim 8, wherein the boundary strength determining module is configured to determine the boundary strength as having a value of 1 based on the first reference frame having a higher resolution than the second frame, and the second frame being the current frame.

19. A computer-readable storage medium storing computer-readable instructions executable by one or more processors, that when executed by the one or more processors, cause the one or more processors to perform operations comprising:

determining block boundaries to be filtered within the current frame;

determining a boundary strength of a block boundary to be filtered based on a difference in resolution between a first reference frame referenced in motion prediction of a block adjacent to the block boundary and a second frame; and

applying a deblocking filter to the block boundary based on the boundary strength

20. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 2 based on the first reference frame having a lower resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

21. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 2

based on the first reference frame having a lower resolution than the second frame, and the second frame being the current frame.

22. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 2 based on the first reference frame having a different resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of another block adjacent to the block boundary.

23. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 1 based on the first reference frame having a lower resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

24. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 1 based on the first reference frame having a lower resolution than the second frame, and the second frame being the current frame.

25. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 2 based on the first reference frame having a higher resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

26. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 2 based on the first reference frame having a higher resolution than the second frame, and the second frame being the current frame.

27. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 2 based on the first reference frame having a different resolution than the second frame, and the second frame being the current frame.

28. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 1 based on the first reference frame having a higher resolution than the second frame, and the second frame being a reference frame referenced in motion prediction of the current frame.

29. The computer-readable storage medium of claim 19, wherein the operations further comprise determining the boundary strength as having a value of 1 based on the first reference frame having a higher resolution than the second frame, and the second frame being the current frame.

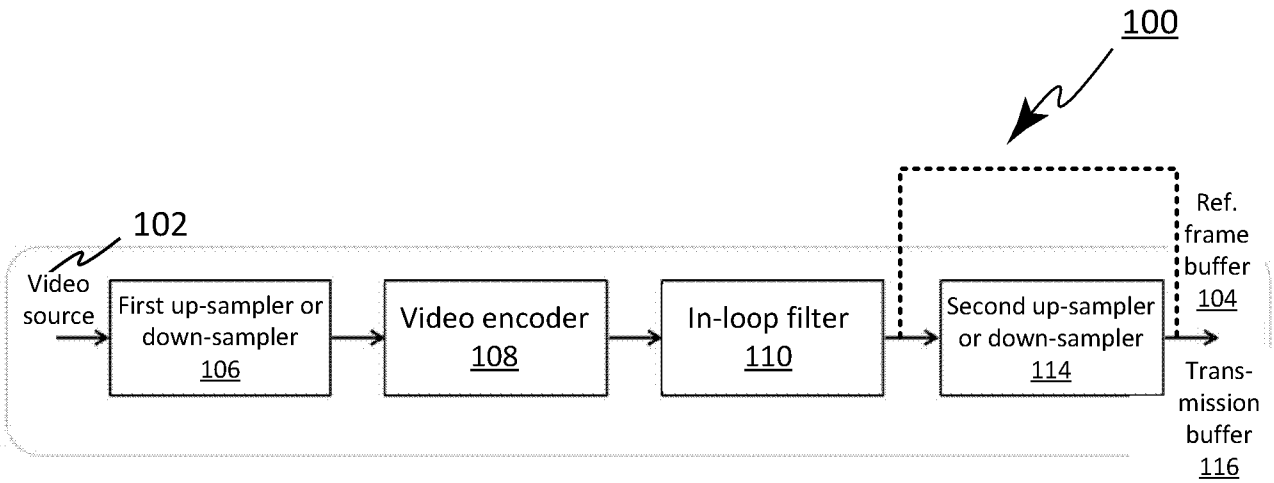
100
102
Video source
First up-sampler or down-sampler 106
Video encoder 108
In-loop filter 110
Second up-sampler or down-sampler 114
Ref. frame buffer 104
Trans-mission buffer 116

FIG. 1A

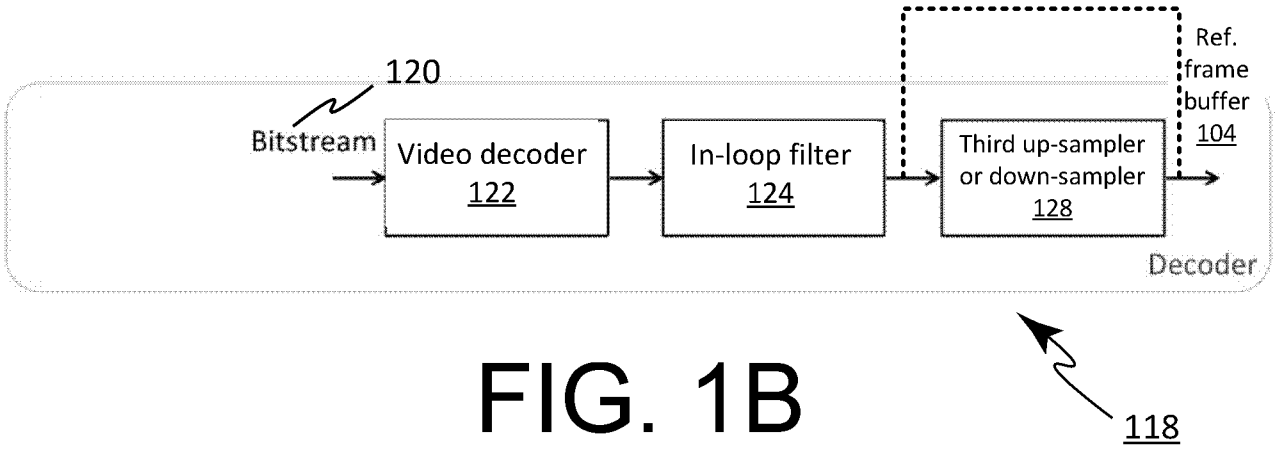
120
Bitstream
Video decoder 122
In-loop filter 124
Third up-sampler or down-sampler 128
Ref. frame buffer 104
Decoder
118

FIG. 1B

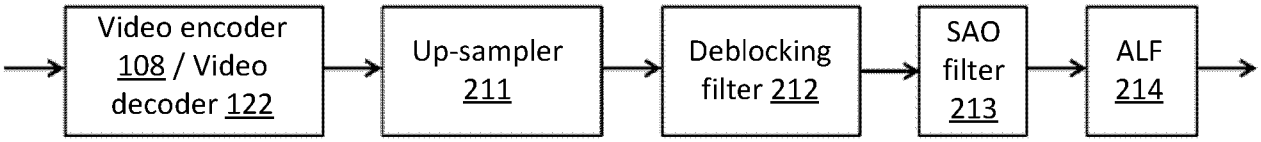

FIG. 2A

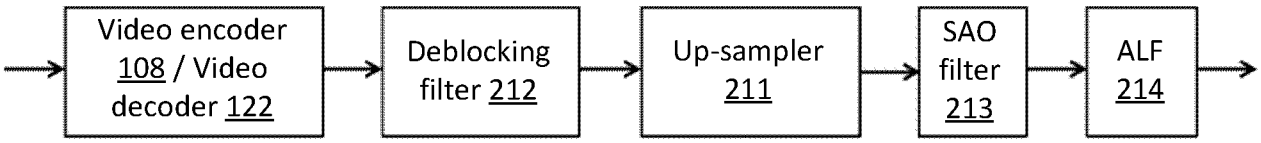

FIG. 2B

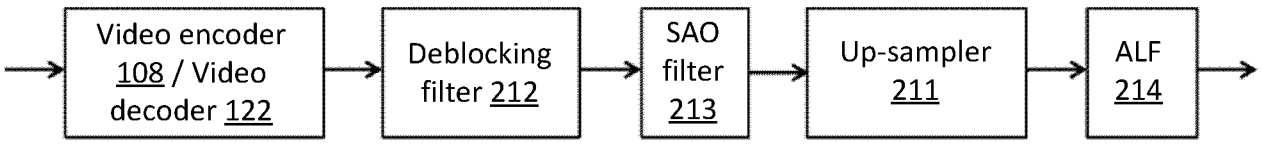

FIG. 2C

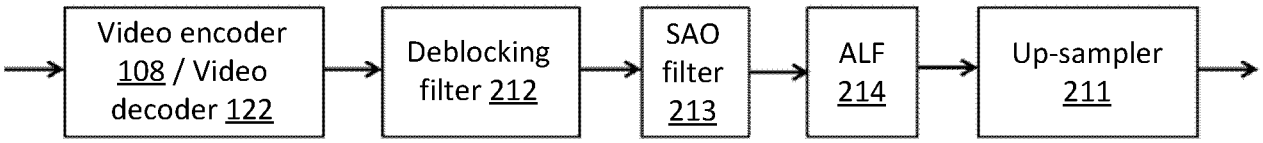

FIG. 2D

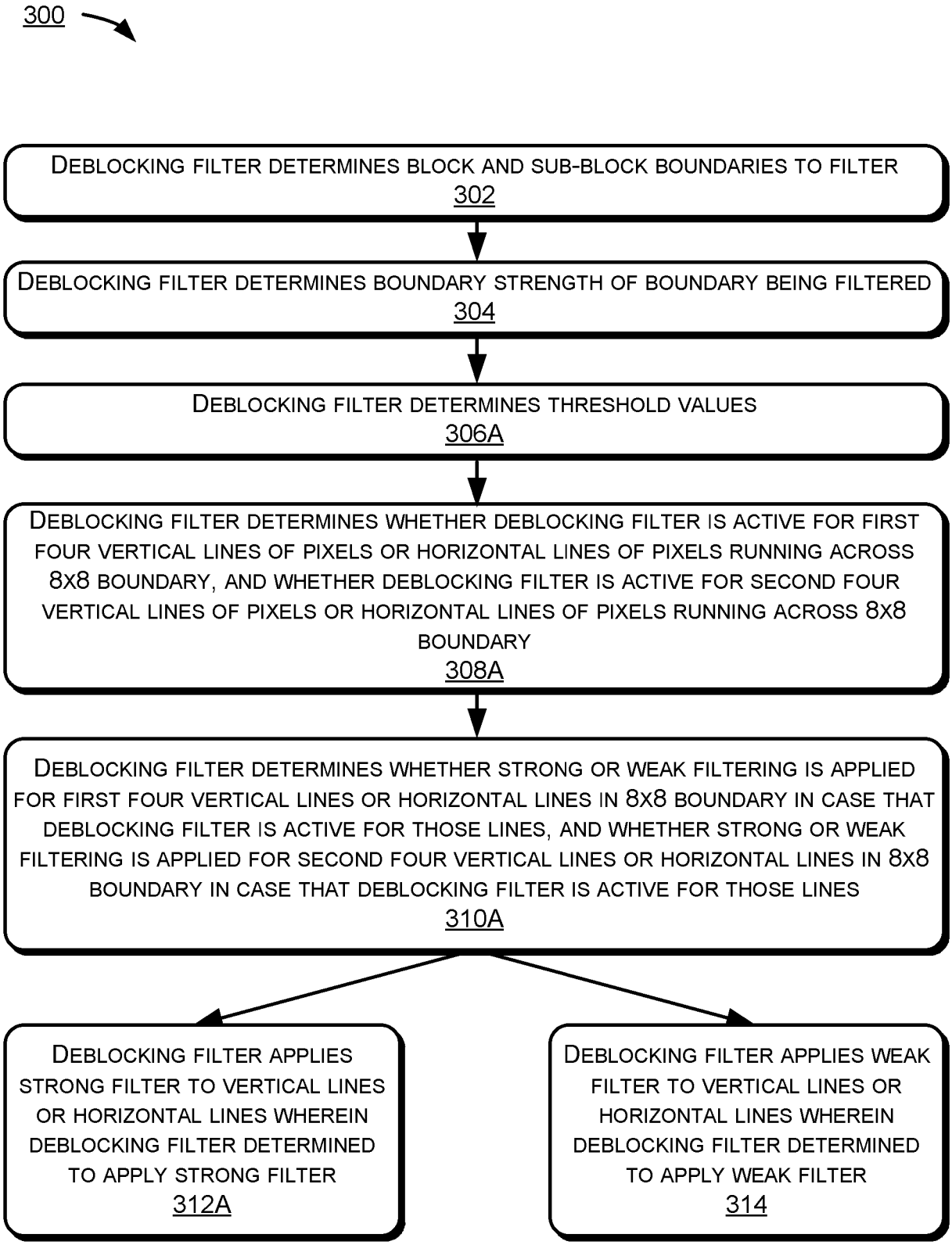
300
DEBLOCKING FILTER DETERMINES BLOCK AND SUB-BLOCK BOUNDARIES TO FILTER
302
DEBLOCKING FILTER DETERMINES BOUNDARY STRENGTH OF BOUNDARY BEING FILTERED
304
DEBLOCKING FILTER DETERMINES THRESHOLD VALUES
306A
DEBLOCKING FILTER DETERMINES WHETHER DEBLOCKING FILTER IS ACTIVE FOR FIRST FOUR VERTICAL LINES OF PIXELS OR HORIZONTAL LINES OF PIXELS RUNNING ACROSS 8X8 BOUNDARY, AND WHETHER DEBLOCKING FILTER IS ACTIVE FOR SECOND FOUR VERTICAL LINES OF PIXELS OR HORIZONTAL LINES OF PIXELS RUNNING ACROSS 8X8 BOUNDARY
308A
DEBLOCKING FILTER DETERMINES WHETHER STRONG OR WEAK FILTERING IS APPLIED FOR FIRST FOUR VERTICAL LINES OR HORIZONTAL LINES IN 8X8 BOUNDARY IN CASE THAT DEBLOCKING FILTER IS ACTIVE FOR THOSE LINES, AND WHETHER STRONG OR WEAK FILTERING IS APPLIED FOR SECOND FOUR VERTICAL LINES OR HORIZONTAL LINES IN 8X8 BOUNDARY IN CASE THAT DEBLOCKING FILTER IS ACTIVE FOR THOSE LINES
310A
DEBLOCKING FILTER APPLIES STRONG FILTER TO VERTICAL LINES OR HORIZONTAL LINES WHEREIN DEBLOCKING FILTER DETERMINED TO APPLY STRONG FILTER
312A
DEBLOCKING FILTER APPLIES WEAK FILTER TO VERTICAL LINES OR HORIZONTAL LINES WHEREIN DEBLOCKING FILTER DETERMINED TO APPLY WEAK FILTER
314

FIG. 3A

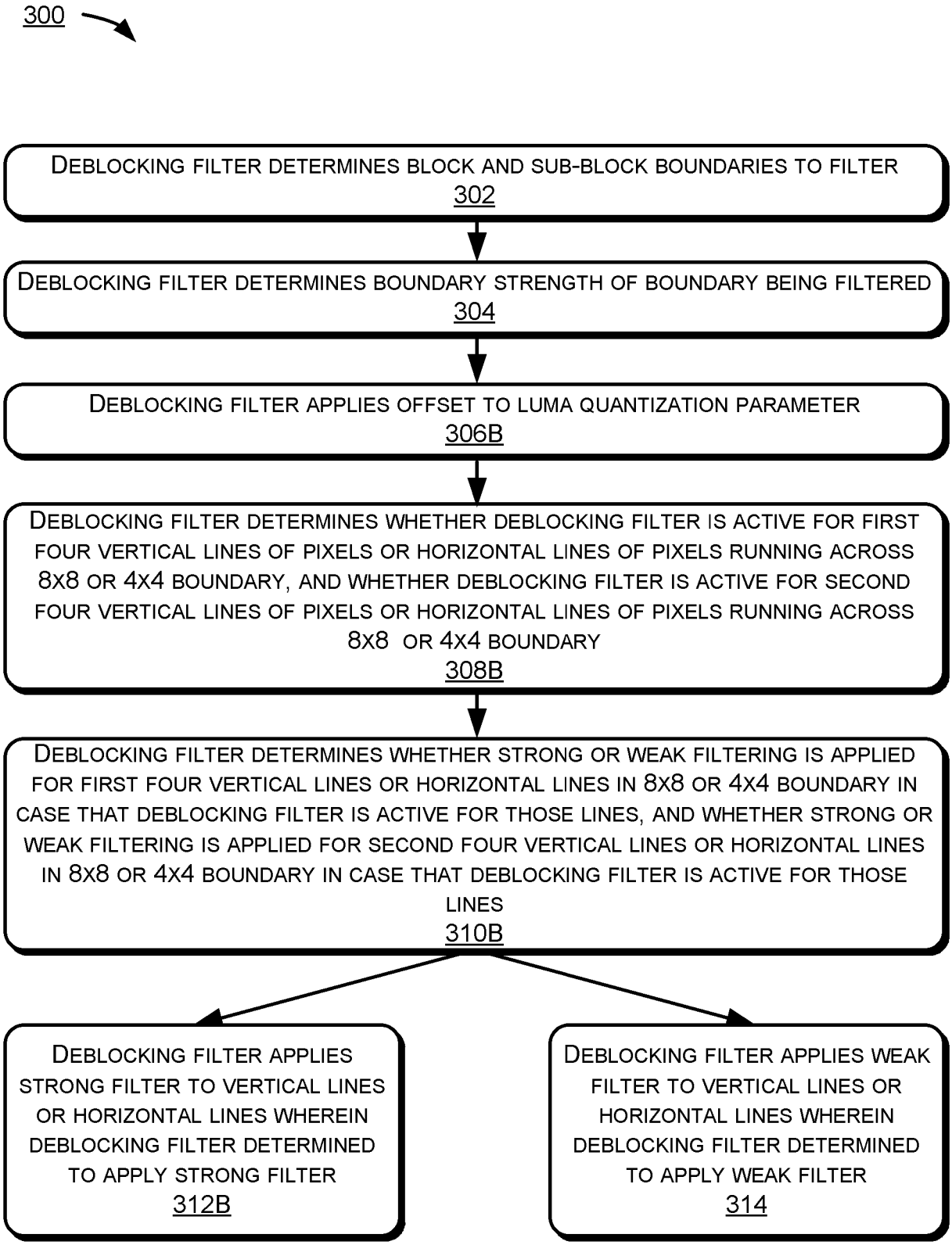
300
DEBLOCKING FILTER DETERMINES BLOCK AND SUB-BLOCK BOUNDARIES TO FILTER
302
DEBLOCKING FILTER DETERMINES BOUNDARY STRENGTH OF BOUNDARY BEING FILTERED
304
DEBLOCKING FILTER APPLIES OFFSET TO LUMA QUANTIZATION PARAMETER
306B
DEBLOCKING FILTER DETERMINES WHETHER DEBLOCKING FILTER IS ACTIVE FOR FIRST FOUR VERTICAL LINES OF PIXELS OR HORIZONTAL LINES OF PIXELS RUNNING ACROSS 8X8 OR 4X4 BOUNDARY, AND WHETHER DEBLOCKING FILTER IS ACTIVE FOR SECOND FOUR VERTICAL LINES OF PIXELS OR HORIZONTAL LINES OF PIXELS RUNNING ACROSS 8X8 OR 4X4 BOUNDARY
308B
DEBLOCKING FILTER DETERMINES WHETHER STRONG OR WEAK FILTERING IS APPLIED FOR FIRST FOUR VERTICAL LINES OR HORIZONTAL LINES IN 8X8 OR 4X4 BOUNDARY IN CASE THAT DEBLOCKING FILTER IS ACTIVE FOR THOSE LINES, AND WHETHER STRONG OR WEAK FILTERING IS APPLIED FOR SECOND FOUR VERTICAL LINES OR HORIZONTAL LINES IN 8X8 OR 4X4 BOUNDARY IN CASE THAT DEBLOCKING FILTER IS ACTIVE FOR THOSE LINES
310B
DEBLOCKING FILTER APPLIES STRONG FILTER TO VERTICAL LINES OR HORIZONTAL LINES WHEREIN DEBLOCKING FILTER DETERMINED TO APPLY STRONG FILTER
312B
DEBLOCKING FILTER APPLIES WEAK FILTER TO VERTICAL LINES OR HORIZONTAL LINES WHEREIN DEBLOCKING FILTER DETERMINED TO APPLY WEAK FILTER
314

FIG. 3B

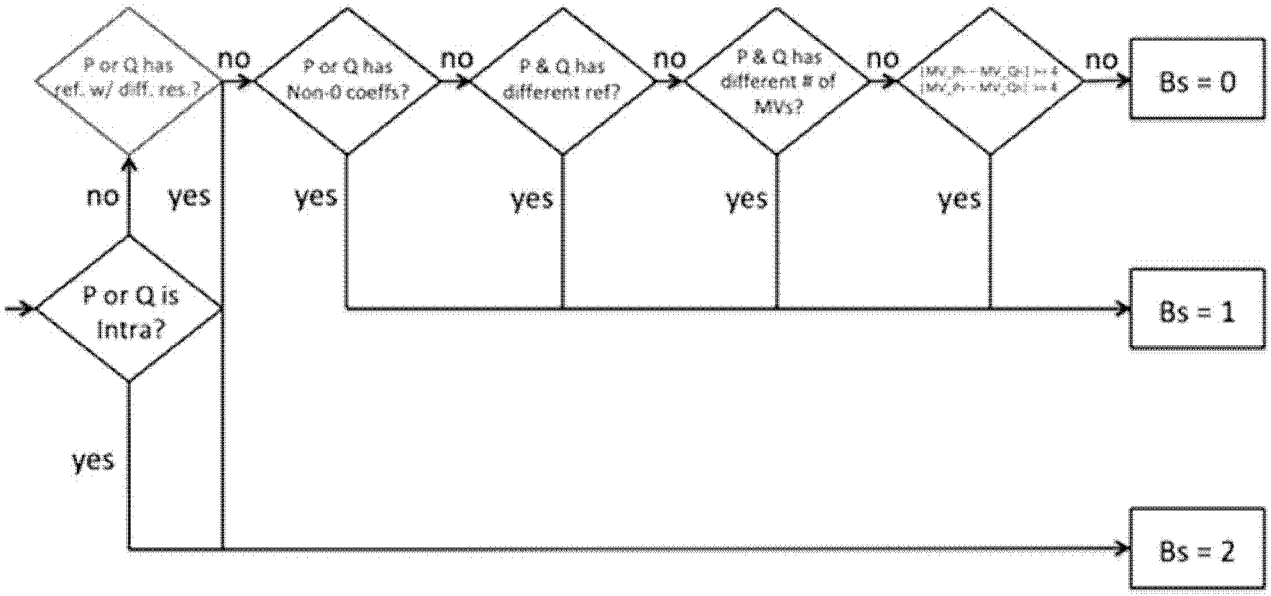
P or Q has ref. w/ diff. res.?
no
P or Q has Non-0 coeffs?
no
P & Q has different ref?
no
P & Q has different # of MVs?
no
|MV_P - MV_Q| >= 4
|MV_P - MV_Q| >= 4
no
Bs = 0
no
yes
yes
yes
yes
yes
P or Q is Intra?
Bs = 1
yes
Bs = 2

FIG. 4A

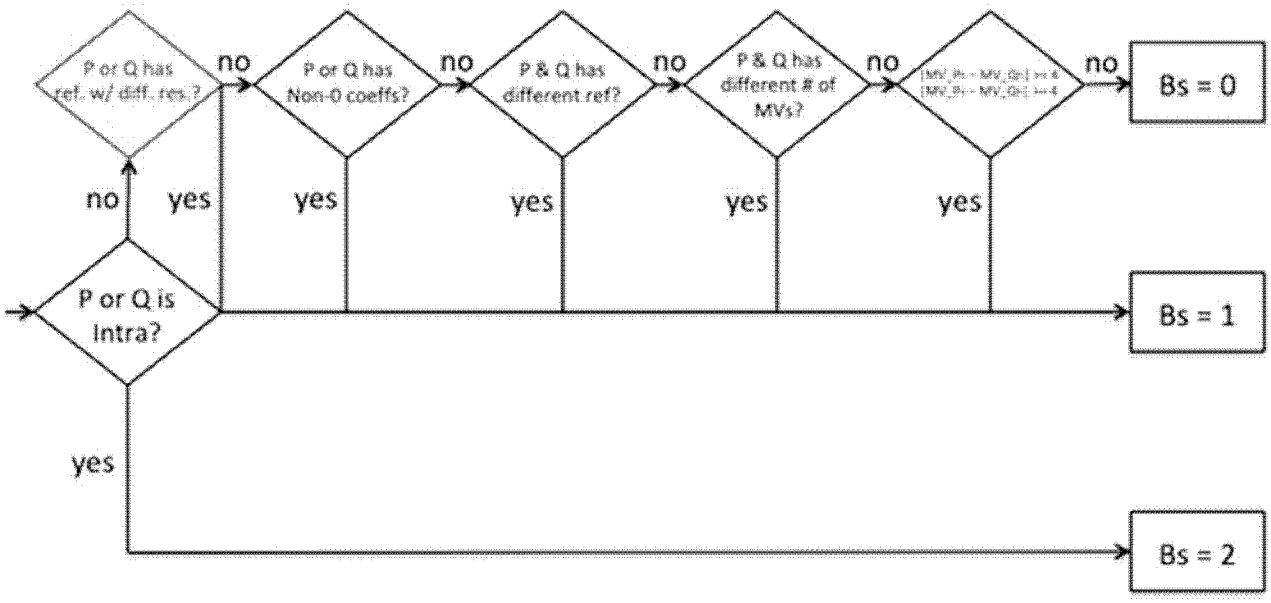
P or Q has ref. w/ diff. res.?
no
P or Q has Non-0 coeffs?
no
P & Q has different ref?
no
P & Q has different # of MVs?
no
|MV_P - MV_Q| >= 4
|MV_P - MV_Q| >= 4
no
Bs = 0
yes
yes
yes
yes
yes
no
P or Q is Intra?
Bs = 1
yes
Bs = 2

FIG. 4B

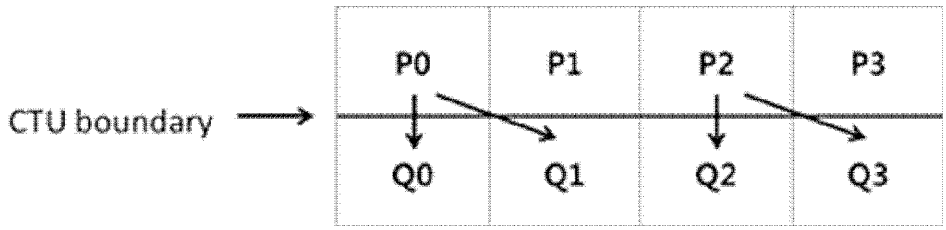

FIG. 5

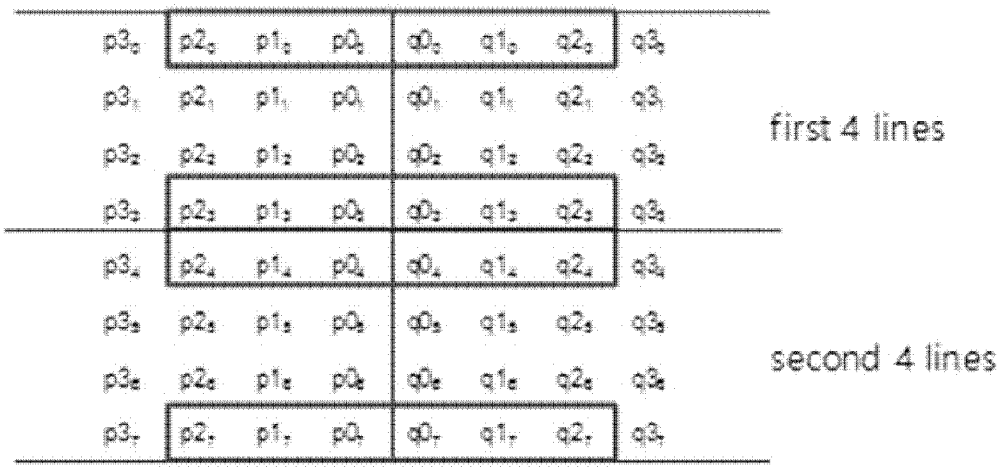

FIG. 6A

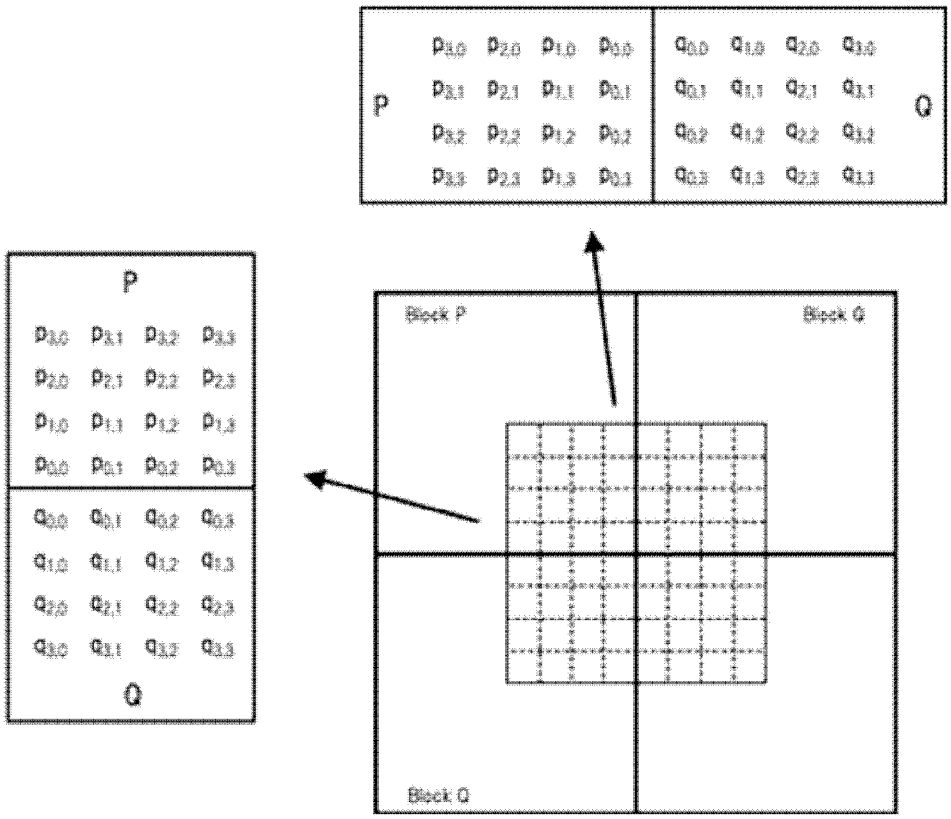

FIG. 6B

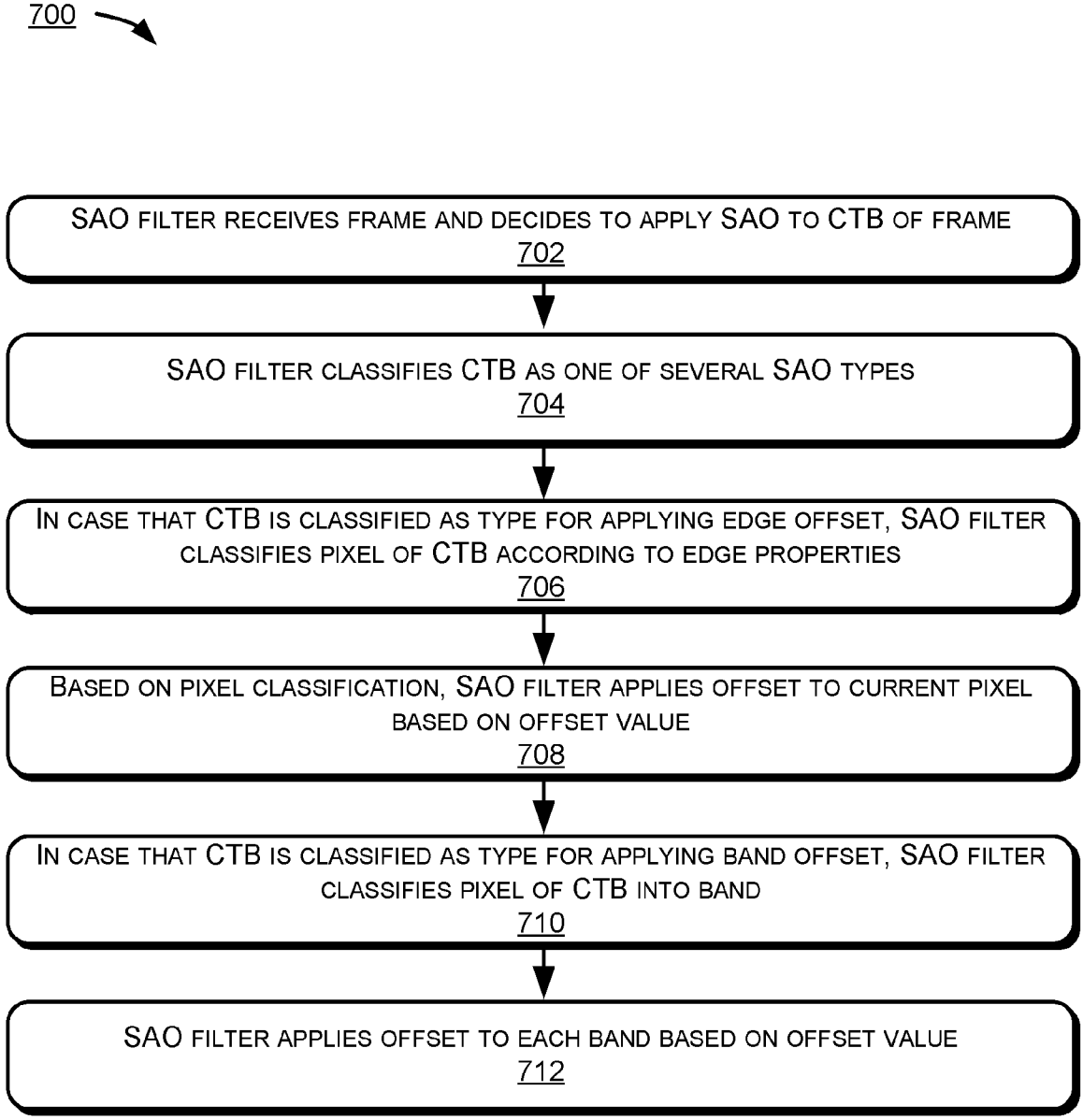
700
SAO FILTER RECEIVES FRAME AND DECIDES TO APPLY SAO TO CTB OF FRAME
702
SAO FILTER CLASSIFIES CTB AS ONE OF SEVERAL SAO TYPES
704
IN CASE THAT CTB IS CLASSIFIED AS TYPE FOR APPLYING EDGE OFFSET, SAO FILTER CLASSIFIES PIXEL OF CTB ACCORDING TO EDGE PROPERTIES
706
BASED ON PIXEL CLASSIFICATION, SAO FILTER APPLIES OFFSET TO CURRENT PIXEL BASED ON OFFSET VALUE
708
IN CASE THAT CTB IS CLASSIFIED AS TYPE FOR APPLYING BAND OFFSET, SAO FILTER CLASSIFIES PIXEL OF CTB INTO BAND
710
SAO FILTER APPLIES OFFSET TO EACH BAND BASED ON OFFSET VALUE
712

FIG. 7

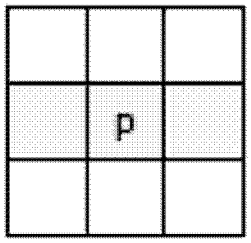

FIG. 8A

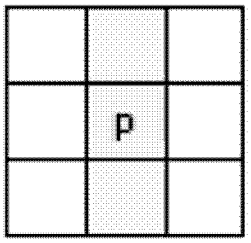

FIG. 8B

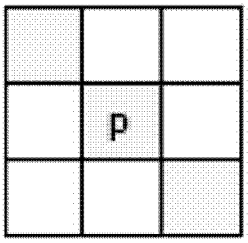

FIG. 8C

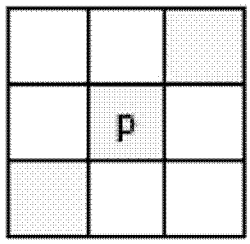

FIG. 8D

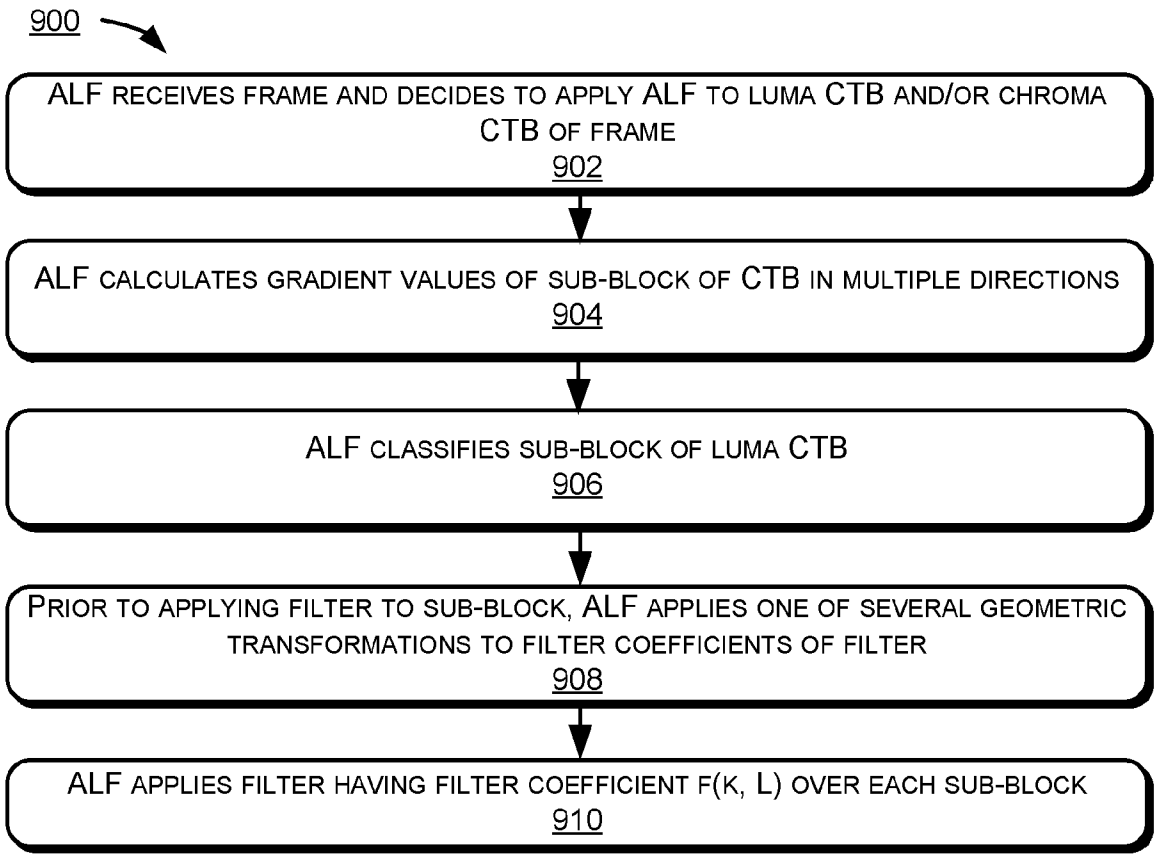
900
ALF RECEIVES FRAME AND DECIDES TO APPLY ALF TO LUMA CTB AND/OR CHROMA CTB OF FRAME
902
ALF CALCULATES GRADIENT VALUES OF SUB-BLOCK OF CTB IN MULTIPLE DIRECTIONS
904
ALF CLASSIFIES SUB-BLOCK OF LUMA CTB
906
PRIOR TO APPLYING FILTER TO SUB-BLOCK, ALF APPLIES ONE OF SEVERAL GEOMETRIC TRANSFORMATIONS TO FILTER COEFFICIENTS OF FILTER
908
ALF APPLIES FILTER HAVING FILTER COEFFICIENT F(K, L) OVER EACH SUB-BLOCK
910

FIG. 9A

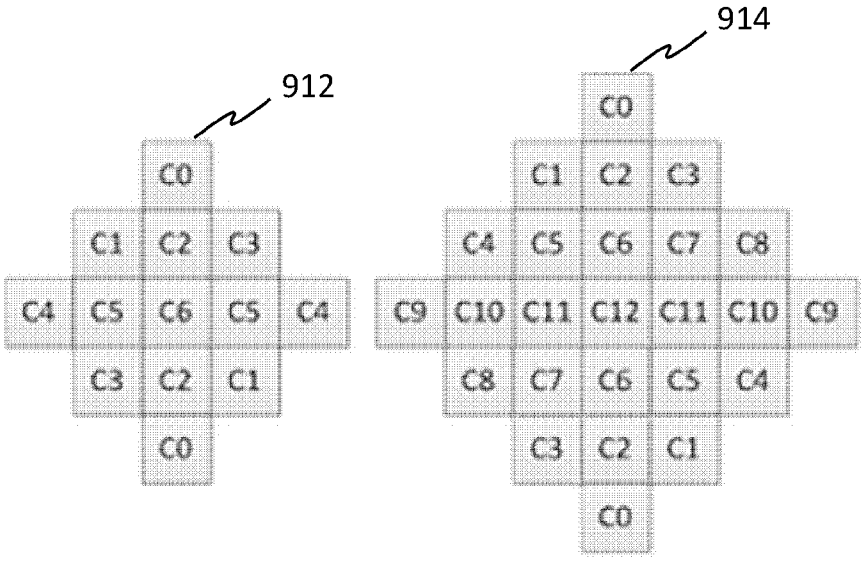
912
C0
C1 C2 C3
C4 C5 C6 C5 C4
C3 C2 C1
C0
914
C0
C1 C2 C3
C4 C5 C6 C7 C8
C9 C10 C11 C12 C11 C10 C9
C8 C7 C6 C5 C4
C3 C2 C1
C0

FIG. 9B

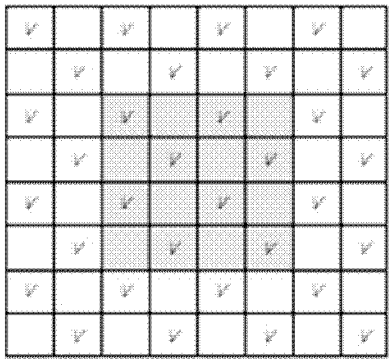

FIG. 9C

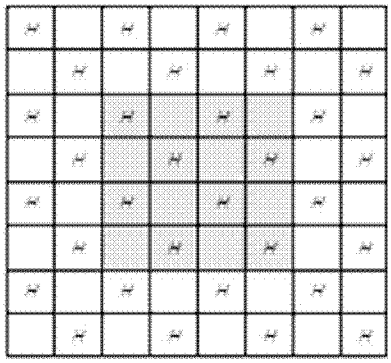

FIG. 9D

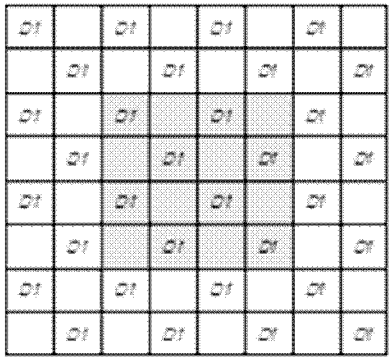

FIG. 9E

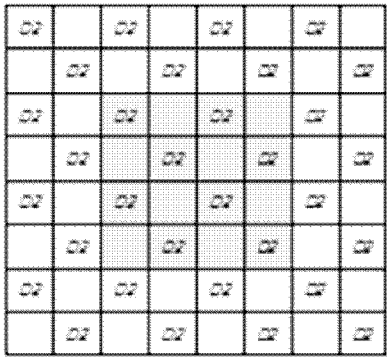

FIG. 9F

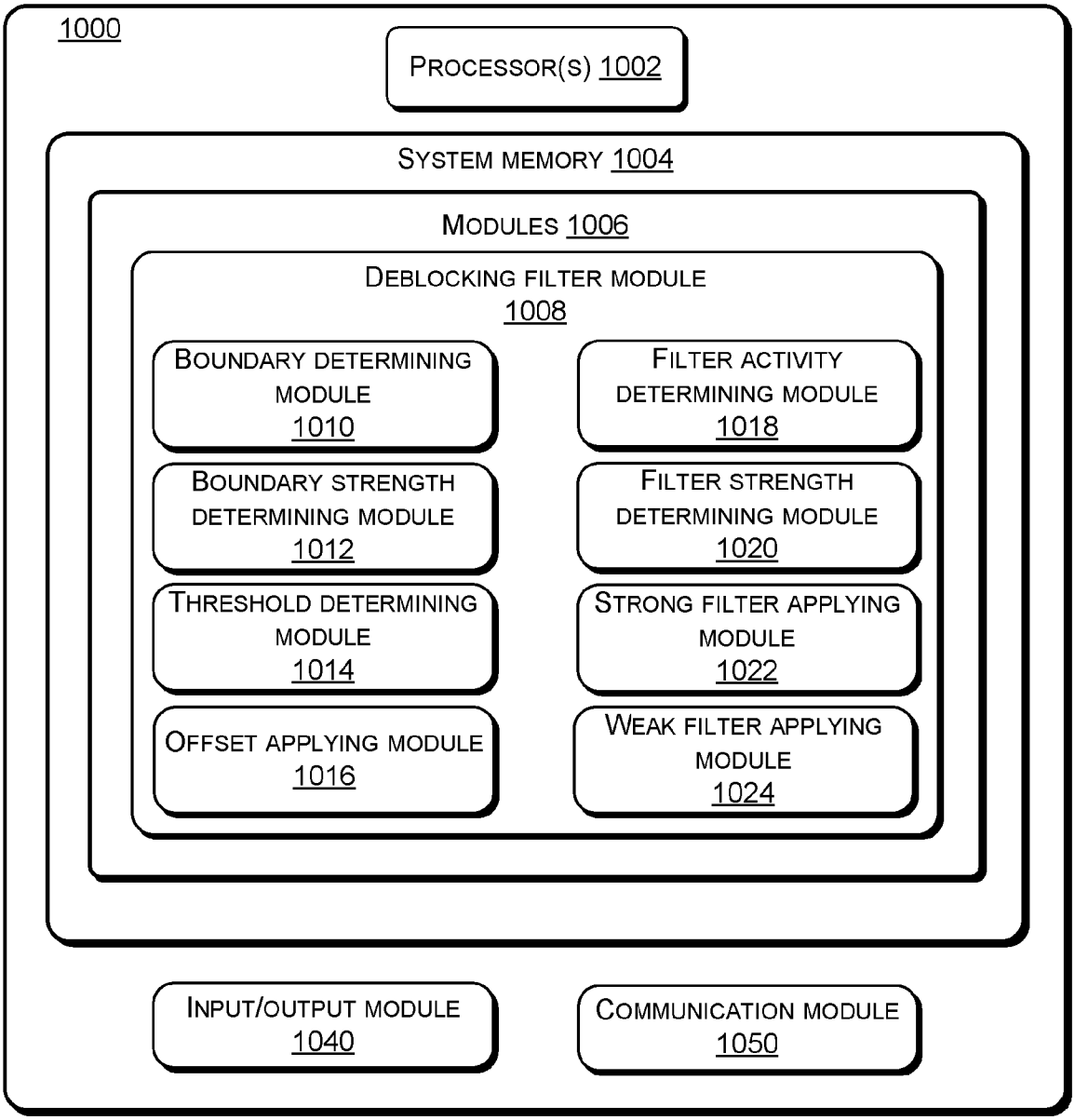
1000
PROCESSOR(S) 1002
SYSTEM MEMORY 1004
MODULES 1006
DEBLOCKING FILTER MODULE 1008
BOUNDARY DETERMINING MODULE 1010
BOUNDARY STRENGTH DETERMINING MODULE 1012
THRESHOLD DETERMINING MODULE 1014
OFFSET APPLYING MODULE 1016
FILTER ACTIVITY DETERMINING MODULE 1018
FILTER STRENGTH DETERMINING MODULE 1020
STRONG FILTER APPLYING MODULE 1022
WEAK FILTER APPLYING MODULE 1024
INPUT/OUTPUT MODULE 1040
COMMUNICATION MODULE 1050

FIG. 10

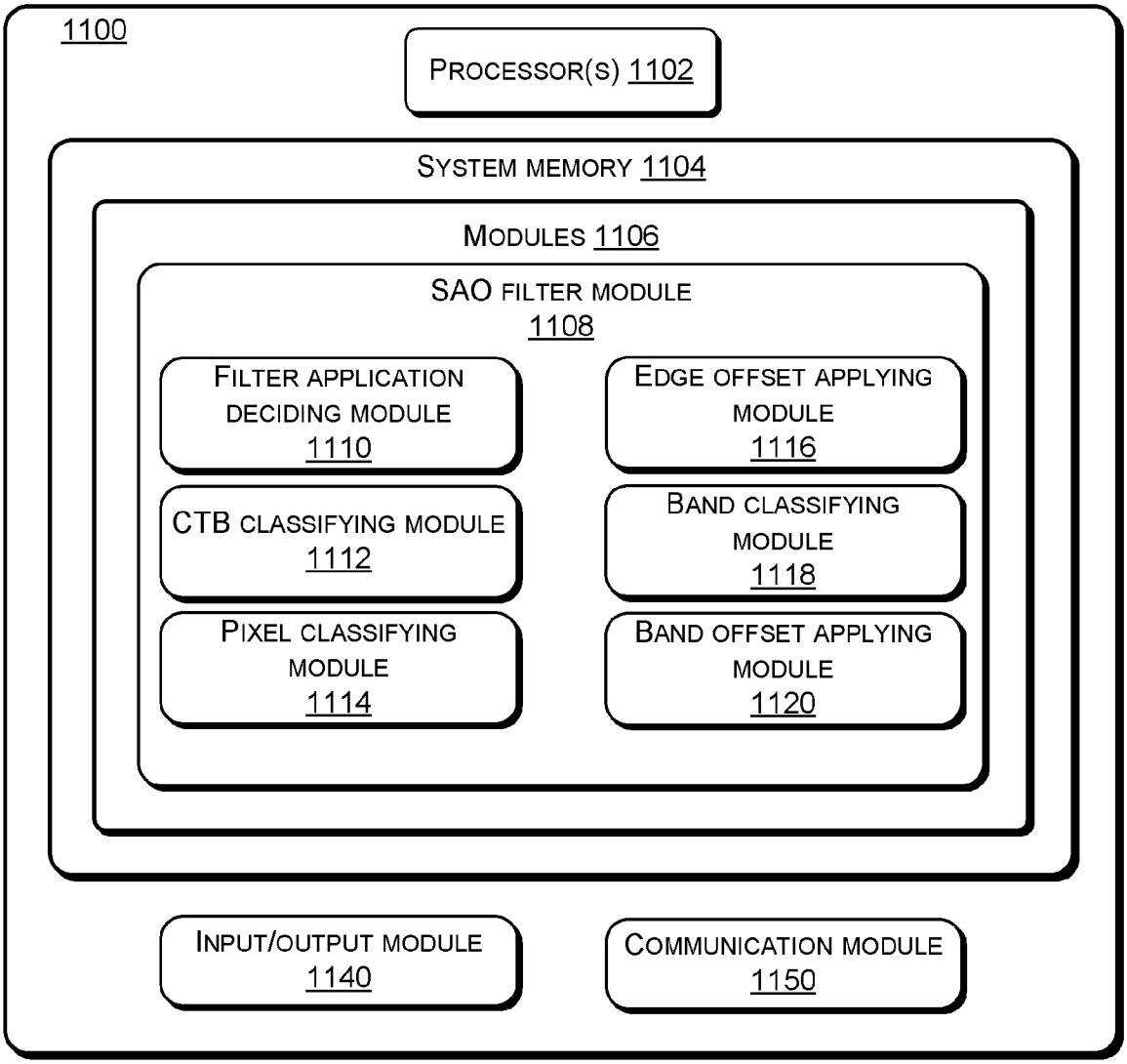
1100
PROCESSOR(S) 1102
SYSTEM MEMORY 1104
MODULES 1106
SAO FILTER MODULE 1108
FILTER APPLICATION DECIDING MODULE 1110
CTB CLASSIFYING MODULE 1112
PIXEL CLASSIFYING MODULE 1114
EDGE OFFSET APPLYING MODULE 1116
BAND CLASSIFYING MODULE 1118
BAND OFFSET APPLYING MODULE 1120
INPUT/OUTPUT MODULE 1140
COMMUNICATION MODULE 1150

FIG. 11

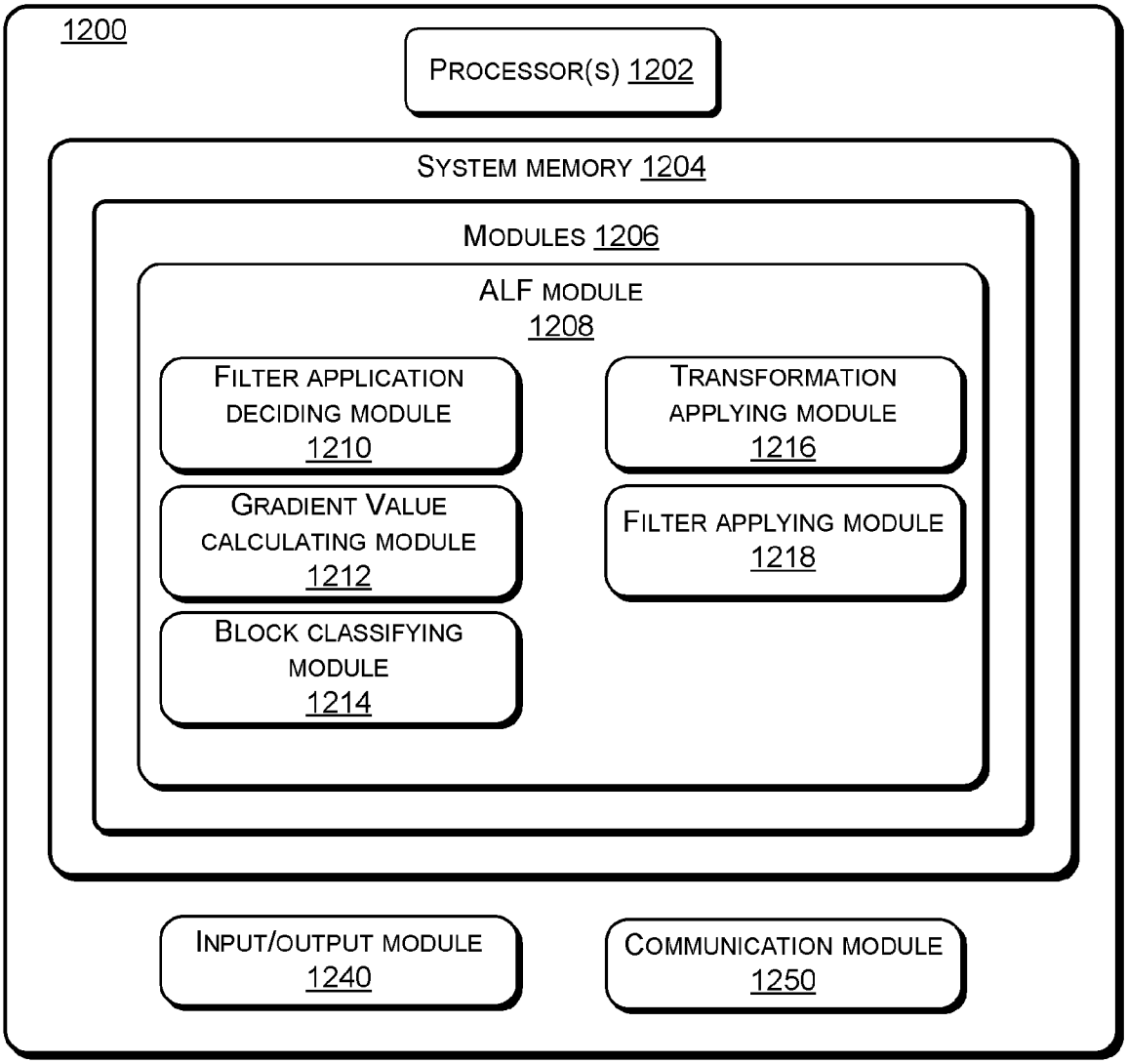
1200
PROCESSOR(S) 1202
SYSTEM MEMORY 1204
MODULES 1206
ALF MODULE 1208
FILTER APPLICATION DECIDING MODULE 1210
GRADIENT VALUE CALCULATING MODULE 1212
BLOCK CLASSIFYING MODULE 1214
TRANSFORMATION APPLYING MODULE 1216
FILTER APPLYING MODULE 1218
INPUT/OUTPUT MODULE 1240
COMMUNICATION MODULE 1250

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/073563

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/117(2014.01)i; H04N 19/86(2014.01)i; H04N 19/82(2014.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNKI, CNPAT, WPI, EPODOC, IEEE, JVET: video, coding, encod+, decod+, frame, block, boundar+, edge, filter+, deblock, de-block+, reference, predict+, estimat+, strength, adaptive, resolution, change, ARC, different, loop, in-loop, strong, weak

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2015365666 A1 (VID SCALE, INC.) 17 December 2015 (2015-12-17) description, paragraphs [0033]-[0149] and figure 11	1-29
A	CN 103931185 A (QUALCOMM INC.) 16 July 2014 (2014-07-16) the whole document	1-29
A	CN 109479152 A (INTERDIGITAL VC HOLDINGS INC.) 15 March 2019 (2019-03-15) the whole document	1-29
A	CN 101267560 A (ZHEJIANG UNIVERSITY) 17 September 2008 (2008-09-17) the whole document	1-29
A	US 2010322304 A1 (NOVATEK MICROELECTRONICS CORP.) 23 December 2010 (2010-12-23) the whole document	1-29
A	WO 2019121164 A1 (TELEFONAKTIEBOLAGET LM ERICSSONPUBL) 27 June 2019 (2019-06-27) the whole document	1-29

☑ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
31 July 2020	**28 August 2020**
Name and mailing address of the ISA/CN **National Intellectual Property Administration, PRC** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088** **China** Facsimile No. **(86-10)62019451**	Authorized officer **WANG,Conglei** Telephone No. **86-(10)-53961717**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/073563

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Wang, Ye-Kui et al. ""On adaptive resolution change (ARC) for VVC"" *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 13th Meeting: Marrakech, MA,* 18 January 2019 (2019-01-18), the whole document	1-29
A	Hong, Seungwook et al. ""AHG19: Adaptive resolution change (ARC) support in VVC"" *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH,* 27 March 2019 (2019-03-27), the whole document	1-29

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/073563

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2015365666	A1	17 December 2015	EP	2941871	A2	11 November 2015
				WO	2014107709	A2	10 July 2014
CN	103931185	A	16 July 2014	HK	1195980	A0	28 November 2014
				SG	11201401379	A1	29 May 2014
				IL	232003	A1	28 May 2014
				ID	201505426	A	04 December 2015
				PH	12014500906	A1	09 June 2014
				MY	166482	A	27 June 2018
				AU	2012328924	A1	08 May 2014
				WO	2013063117	A1	02 May 2013
				US	2013101024	A1	25 April 2013
				TW	201334544	A	16 August 2013
				KR	20140085545	A	07 July 2014
				EP	2772053	A1	03 September 2014
				JP	2014534733	A	18 December 2014
				HK	1195980	A1	27 April 2018
				VN	39234	A	25 September 2014
				RU	2014121089	A	10 December 2015
				CA	2852533	A1	02 May 2013
				BR	112014009431	A2	18 April 2017
CN	109479152	A	15 March 2019	US	2019200045	A1	27 June 2019
				KR	20190029523	A	20 March 2019
				JP	2019519132	A	04 July 2019
				EP	3244617	A1	15 November 2017
				WO	2017194297	A1	16 November 2017
				EP	3456055	A1	20 March 2019
CN	101267560	A	17 September 2008	None			
US	2010322304	A1	23 December 2010	TW	201101838	A	01 January 2011
WO	2019121164	A1	27 June 2019	None			